

**PENGEMBANGAN SISTEM INFORMASI ASPIRASI *ONLINE*
BERBASIS WEB MENGGUNAKAN PEMODELAN *REUSE-
ORIENTED DEVELOPMENT*
(STUDI KASUS : DPM UNIVERSITAS BRAWIJAYA)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Ali Fikri
NIM: 145150407111074



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI ASPIRASI *ONLINE* BERBASIS WEB
MENGUNAKAN PEMODELAN *REUSE-ORIENTED DEVELOPMENT*
(STUDI KASUS : DPM UNIVERSITAS BRAWIJAYA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ali Fikri

NIM: 145150407111074

Skripsi ini telah diuji dan dinyatakan lulus pada
18 Oktober 2018

Telah diperiksa dan disetujui oleh:

Pembimbing I

Ismiarta Aknuranda, S.T, M.Sc, Ph.D.

NIK: 201006 740719 1 001

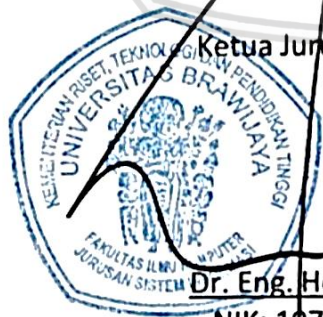
Pembimbing II

Fajar Pradana, S.ST, M.Eng.

NIP: 19871121 201504 1 004

Mengetahui

Ketua Jurusan Sistem Informasi



Dr. Eng. Herman Tolle, S.T., M.T.

NIK: 19740823 200012 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 19 Oktober 2018



Ali Fikri

NIM: 145150407111074

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pengembangan Sistem Informasi Aspirasi *Online* Berbasis Web Menggunakan Pemodelan *Reuse-Oriented Development* (Studi Kasus : DPM Universitas Brawijaya)” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibu saya, Hj. Nur Rahmah dan ayah saya, H. M. Atrawi S.Pd.I, yang selalu memberikan dukungan do’a, moril, materil, motivasi, dan hal lain yang senantiasa menjadi contoh yang baik sehingga penulis dapat menyelesaikan skripsi ini. Dan tak lupa juga buat adik-adik saya, Safinatunnajatirrahmah (Fina) dan Arum Hayatal Magfirah (Fira) yang selalu menghadirkan rasa kangen, canda dan tawa dalam masa-masa jauh dari rumah.
2. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D. selaku pembimbing I, serta Bapak Fajar Pradana, S.ST, M.Eng. selaku pembimbing II yang telah sabar dan tulus dalam memberikan banyak masukan, saran, arahan dan ilmu selama proses pengerjaan skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku dekan Fakultas Ilmu Komputer Universitas Brawijaya, Bapak Dr. Eng. Herman Tolle, S.T., M.MT., selaku ketua jurusan Sistem Informasi dan Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku Ketua Program Studi Sistem Informasi.
4. Farhan Caesarian dan Anggota DPM UB lainnya yang dengan senang hati membagikan informasi, permasalahan yang ada dan memberikan kesempatan kepada saya melakukan penelitian pada DPM UB.
5. Teman-teman di BEM FILKOM Sukmakarya, teman-teman di Kementrian Karya Prestatif, teman-teman ospek kelompok 9 WOW SIUB 2014, teman-teman yang tergabung dalam grup Mak Tin S.Kom., grup Jihad Fisabilillah dan teman-teman angkatan SIUB2014 yang tak bisa disebutkan namanya satu-persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 18 Oktober 2018

Ali Fikri

afikri124@gmail.com

ABSTRAK

Ali Fikri, Pengembangan Sistem Informasi Aspirasi *Online* Berbasis Web Menggunakan Pemodelan *Reuse-Oriented Development* (Studi Kasus : DPM Universitas Brawijaya)

Pembimbing: Ismiarta Aknuranda, S.T, M.Sc, Ph.D. dan Fajar Pradana, S.ST, M.Eng.

Dewan Perwakilan Mahasiswa Universitas Brawijaya (DPM UB) merupakan lembaga legislatif yang mempunyai fungsi advokasi termasuk dalam hal menyangkut aspirasi mahasiswa maupun lembaga lainnya. Namun, dalam pelayanannya masih terkendala dalam proses penyampaian dan penyerapannya. Akibatnya, berdampak pada kurang optimalnya kinerja organisasi kemahasiswaan yang berada dalam struktur Lembaga Kedaulatan Mahasiswa Universitas Brawijaya (LKM UB). Untuk mengoptimalkannya, maka diperlukan implementasi *good student governance* pada DPM UB dengan pemanfaatan teknologi informasi sebagai instrumen yang digunakan untuk meningkatkan layanan advokasi pada lembaga tersebut.

Berdasarkan permasalahan tersebut, perlu dikembangkan suatu sistem informasi yang menangani aspirasi secara *online* berbasis web pada DPM UB untuk mempermudah dan mempercepat pelayanan yang diberikan. Metode yang digunakan dalam pengembangannya adalah *reuse-oriented development model* yang memiliki beberapa keuntungan dalam proses penerapannya. Untuk memvalidasi persyaratan fungsional sistem maka dilakukan pengujian validasi dengan metode *black-box*, pengujian kompatibilitas untuk mengetahui *compatibility* dan *accessibility* sistem, pengujian penerimaan pengguna untuk mengetahui persentase penerimaan pengguna terhadap *usability* dan *security* sistem yang dikembangkan, dan pengujian efisiensi waktu untuk mengetahui efisiensi waktu sebelum dan sesudah adanya sistem.

Hasil dari penelitian ini berupa Sistem Informasi Aspirasi *Online* pada DPM UB beserta hasil uji coba penerapan sistem tersebut. Hasil dari uji validasi yang dilakukan didapatkan sebesar 100% valid, hasil uji kompatibilitas mendapatkan skor rata-rata 89%, hasil uji penerimaan pengguna mendapatkan skor rata-rata keseluruhan sebesar 81%, dan hasil uji efisiensi waktu sebesar 84,4%. Sehingga dapat disimpulkan bahwa pemanfaatan teknologi yang digunakan dapat diterima dan memberikan dampak positif pada pelayanan advokasi di kelembagaan LKM UB.

Kata kunci: Aspirasi, DPM, LKM, Pengembangan Sistem Informasi, *Reuse-oriented*.

ABSTRACT

Ali Fikri, Development of Web-based Online Aspiration Information System Using Reuse-Oriented Development Modelling (Case Study : DPM University of Brawijaya)

Mentors: Ismiarta Aknuranda, S.T., M.Sc., Ph.D. and Fajar Pradana, S.ST., M.Eng.

Student Representative Council University of Brawijaya (DPM UB) is a legislative institution that has an advocate function in terms of collecting student aspirations and other institutions. However, the service has some problems in the process of submission and its operation. As a result, the performance of student organizations in the structure in the structure of the Student Sovereignty Institution University of Brawijaya (LKM UB) is less optimal. In order to optimize the system, it is necessary to implement a good student governance at DPM UB by utilizing information technology as an instrument used to improve advocacy services at these institutions.

Based on these problems, it is necessary to develop an information system that handles web-based online aspirations at DPM UB to facilitate and accelerate the services provided. The method used in its development is a reuse-oriented development model which has several advantages in the process of its application. In order to validate the functional system, validation testing with black-box method is used in this case, compatibility testing is used to determine the system compatibility and accessbility, user acceptance testing is used to determine the percentage of user acceptance of usability and security system that's being developed, and time efficiency testing is used to find out before and after used the system

The results of this study is an online aspiration information system at DPM UB along with the results from the implementation test of the system. The results of the validation test were obtained at 100% valid, the compatibility test got an average score 89%, the results of user acceptance tests got an overall average score of 81%, and the results of time efficiency testing got an score 84,4%. It can be concluded that the usage of the implemented technology can be accepted and have a positive impact on advocacy services in the institutions of LKM UB.

Keywords: Aspiration, DPM, LKM, Information System Development, Reuse-oriented.

DAFTAR ISI

HALAMAN SAMPUL.....	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	3
1.3 Tujuan	3
1.4 Manfaat.....	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Tinjauan Pustaka	6
2.2 Gambaran Organisasi.....	6
2.2.1 Profil Organisasi	6
2.2.2 Visi Organisasi	7
2.2.3 Misi Organisasi	7
2.2.4 Tugas dan Wewenang	7
2.2.5 Lokasi Organisasi	7
2.2.6 Struktur Organisasi.....	7
2.3 Definisi Aspirasi <i>Online</i>	8
2.3.1 Aspirasi	8
2.3.2 <i>Online</i>	9

2.4 Customer Relationship Management (CRM)	9
2.4.1 Jenis-Jenis CRM	10
2.4.2 CRM Operational	10
2.4.3 Complaint Management	10
2.5 Rekayasa Perangkat Lunak	11
2.6 Software Process Model	11
2.6.1 Reuse-Oriented Development	12
2.7 Pemodelan Proses Bisnis	15
2.7.1 Analisis dan Pemodelan Proses Bisnis <i>As-Is</i>	15
2.7.2 Analisis dan Pemodelan Proses Bisnis <i>To-Be</i>	15
2.7.3 Business Process Modeling Notation (BPMN)	16
2.8 Analisis Persyaratan	17
2.8.1 Identifikasi Pemangku Kepentingan	18
2.8.2 Identifikasi Permasalahan Pemangku Kepentingan	19
2.8.3 Identifikasi Kebutuhan Pemangku Kepentingan	19
2.8.4 Identifikasi Fitur	19
2.9 Unified Modelling Language (UML)	20
2.9.1 Use Case Diagram	20
2.9.2 Sequence Diagram	23
2.9.3 Class Diagram	23
2.10 Model View Controller (MVC)	24
2.11 Kuesioner	25
2.12 Pengujian	26
2.12.1 Pengujian <i>Black Box</i>	26
2.12.2 Pengujian Validasi	26
2.12.3 Pengujian Kompatibilitas	27
2.12.4 Pengujian Penerimaan Pengguna	27
2.12.5 Pengujian Efisiensi Waktu	29
BAB 3 METODOLOGI PENELITIAN	30
3.1 Studi Literatur	30
3.2 Pemodelan Proses Bisnis	31
3.3 Analisis Persyaratan	31

3.4 Analisis Komponen.....	31
3.5 Modifikasi Persyaratan	32
3.6 Perancangan dengan Komponen.....	32
3.7 Implementasi	32
3.8 Pengujian dan Analisis	32
3.9 Penulisan Laporan.....	33
BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN	34
4.1 Pemodelan Proses Bisnis	34
4.1.1 Proses Bisnis <i>As-Is</i>	34
4.1.2 Proses Bisnis <i>To-Be</i>	38
4.2 Analisis Persyaratan	42
4.2.1 Elisitasi Persyaratan	42
4.2.2 Spesifikasi persyaratan.....	50
4.2.3 Verifikasi dan Validasi	57
4.3 Pemodelan <i>Use Case</i>	59
4.3.1 Identifikasi Aktor	59
4.3.2 Identifikasi <i>Use Case</i>	60
4.3.3 Pemodelan Diagram <i>Use Case</i>	61
4.3.4 Spesifikasi <i>Use Case</i>	61
BAB 5 ANALISIS KOMPONEN DAN PERANCANGAN	75
5.1 Analisis Komponen.....	75
5.2 Modifikasi Persyaratan	77
5.3 Perancangan dengan Komponen.....	78
5.3.1 Perancangan Arsitektur Sistem	78
5.3.2 Perancangan Interaksi Objek	78
5.3.3 Perancangan Kelas Objek	84
5.3.4 Perancangan Basis Data	87
5.3.5 Perancangan dan Integrasi Komponen	91
5.3.6 Perancangan Antarmuka.....	95
BAB 6 IMPLEMENTASI	100
6.1 Spesifikasi Lingkungan Implementasi	100
6.2 Implementasi dan Integrasi Algoritma	101

6.2.1 Implementasi Fungsi Login Mahasiswa	101
6.2.2 Implementasi Fungsi Login Lembaga	103
6.2.3 Implementasi Fungsi Mengirimkan Aspirasi	104
6.2.4 Implementasi Fungsi Memantau Aspirasi.....	106
6.2.5 Implementasi Fungsi Memberikan Tanggapan.....	108
6.2.6 Implementasi Fungsi Mengelola Akun.....	109
6.3 Implementasi Antarmuka Pengguna	111
6.3.1 Implementasi Antarmuka Halaman Utama	111
6.3.2 Implementasi Antarmuka Login Admin	111
6.3.3 Implementasi Antarmuka Profil	112
6.3.4 Implementasi Antarmuka Kirim Aspirasi	112
6.3.5 Implementasi Antarmuka Statistik Aspirasi	113
6.3.6 Implementasi Antarmuka Kelola Data Aspirasi	113
6.3.7 Implementasi Antarmuka Detail Aspirasi	114
6.3.8 Implementasi Antarmuka Cetak Aspirasi.....	114
6.3.9 Implementasi Antarmuka Kelola Akun	115
BAB 7 PENGUJIAN DAN ANALISIS.....	116
7.1 Pengujian Validasi	116
7.1.1 Rencana Pengujian Validasi	116
7.1.2 Hasil Pengujian Validasi.....	117
7.1.3 Analisis Hasil Pengujian Validasi	122
7.2 Pengujian Kompatibilitas	123
7.2.1 Hasil Pengujian Kompatibilitas.....	123
7.2.2 Analisis Hasil Pengujian Kompatibilitas.....	123
7.3 Pengujian Penerimaan Pengguna	124
7.3.1 Hasil Pengujian Penerimaan Pengguna.....	127
7.3.2 Analisis Hasil Pengujian Penerimaan Pengguna	128
7.4 Pengujian Efisiensi Waktu.....	133
7.4.1 Hasil Pengujian Efisiensi Waktu	133
7.4.2 Analisis Hasil Pengujian Efisiensi Waktu	136
BAB 8 PENUTUP	139
8.1 Kesimpulan.....	139

8.2 Saran	140
DAFTAR PUSTAKA.....	141
LAMPIRAN A HASIL WAWANCARA.....	143
LAMPIRAN B VERIFIKASI DAN VALIDASI PERSYARATAN	145
LAMPIRAN C KUESIONER PENGUJIAN PENERIMAAN PENGGUNA	146



DAFTAR TABEL

Tabel 2.1 Simbol Notasi pada BPMN	16
Tabel 2.2 Kerangka Dokumentasi Pernyataan Masalah	19
Tabel 2.3 Contoh Tabel Fitur	19
Tabel 2.4 Simbol <i>Use Case Diagram</i>	20
Tabel 2.5 Spesifikasi <i>Use Case</i>	22
Tabel 2.6 Simbol <i>Sequence Diagram</i>	23
Tabel 2.7 Simbol <i>Class Diagram</i>	24
Tabel 4.1 Kelebihan Kekurangan Proses Bisnis Saat Ini (<i>as-is</i>)	38
Tabel 4.2 Kelebihan Kekurangan Proses Bisnis Usulan (<i>to-be</i>)	40
Tabel 4.3 Perubahan aktivitas proses bisnis pelayanan apirasi secara umum	41
Tabel 4.4 Hasil identifikasi pemangku kepentingan	42
Tabel 4.5 Pernyataan masalah pertama	43
Tabel 4.6 Pernyataan masalah kedua	44
Tabel 4.7 Identifikasi kebutuhan pemangku kepentingan	45
Tabel 4.8 Hasil Identifikasi Pengguna	47
Tabel 4.9 Hasil Identifikasi Fitur	48
Tabel 4.10 Hubungan kebutuhan pemangku kepentingan dengan fitur	49
Tabel 4.11 Persyaratan Fungsional	51
Tabel 4.12 Persyaratan Non Fungsional	57
Tabel 4.13 Verifikasi dan Validasi Persyaratan Fungsional	58
Tabel 4.14 Verifikasi dan Validasi Persyaratan Non Fungsional	58
Tabel 4.15 Identifikasi Aktor	59
Tabel 4.16 Hubungan <i>Use Case</i> dengan Persyaratan Fungsional	60
Tabel 4.17 Spesifikasi <i>Use Case Login</i>	62
Tabel 4.18 Spesifikasi <i>Use Case Logout</i>	63
Tabel 4.19 Spesifikasi <i>Use Case</i> Memperbarui Identitas	64
Tabel 4.20 Spesifikasi <i>Use Case</i> Mengirimkan Aspirasi	65
Tabel 4.21 Spesifikasi <i>Use Case</i> Mengelola Aspirasi	66
Tabel 4.22 Spesifikasi <i>Use Case</i> Melihat Statistik Aspirasi	68
Tabel 4.23 Spesifikasi <i>Use Case</i> Memberikan Tanggapan	69

Tabel 4.24 Spesifikasi <i>Use Case</i> Memantau Aspirasi	70
Tabel 4.25 Spesifikasi <i>Use Case</i> Mengelola Akun	71
Tabel 4.26 Spesifikasi <i>Use Case</i> Menghapus Aspirasi	73
Tabel 5.1 Daftar komponen yang dapat digunakan ulang	75
Tabel 5.2 Daftar Modifikasi Komponen	77
Tabel 5.3 Struktur Tabel siap_aspirasi	88
Tabel 5.4 Struktur Tabel siap_tanggapan	88
Tabel 5.5 Struktur Tabel siap_notifikasi	88
Tabel 5.6 Struktur Tabel siap_user	89
Tabel 5.7 Struktur Tabel siap_mahasiswa	89
Tabel 5.8 Struktur Tabel siap_lembaga	89
Tabel 5.9 Struktur Tabel siap_admin	89
Tabel 5.10 Struktur Tabel siap_penerima	90
Tabel 5.11 Struktur Tabel siap_status	90
Tabel 5.12 Struktur Tabel siap_kategori	90
Tabel 5.13 Struktur Tabel siap_post	91
Tabel 5.14 Perancangan Komponen Fungsi Login Mahasiswa	91
Tabel 5.15 Perancangan Komponen Fungsi Otentikasi BAIS	92
Tabel 5.16 Perancangan Komponen Fungsi Login Lembaga	92
Tabel 5.17 Perancangan Komponen Fungsi Mengirimkan Aspirasi	93
Tabel 5.18 Perancangan Komponen Fungsi Memantau Aspirasi	93
Tabel 5.19 Perancangan Komponen Fungsi Memberikan tanggapan	94
Tabel 5.20 Perancangan Komponen Fungsi Mengelola Akun	94
Tabel 6.1 Spesifikasi Perangkat Keras	100
Tabel 6.2 Spesifikasi Perangkat Lunak	100
Tabel 6.3 Implementasi Kelas Controller Mahasiswa	101
Tabel 6.4 Implementasi Kelas Model REST_API	103
Tabel 6.5 Implementasi Kelas Controller Lembaga	104
Tabel 6.6 Implementasi Fungsi Mengirimkan Aspirasi	104
Tabel 6.7 Implementasi Fungsi Memantau Aspirasi	106
Tabel 6.8 Implementasi Fungsi Time Ago Helpers	108
Tabel 6.9 Implementasi Fungsi Mengirimkan Aspirasi	108

Tabel 6.10 Implementasi Fungsi Mengelola Akun	109
Tabel 7.1 Rencana Pengujian Validasi	116
Tabel 7.2 Kasus Uji Validasi Login	117
Tabel 7.3 Kasus Uji Validasi logout	117
Tabel 7.4 Kasus Uji Validasi Memperbarui Identitas	118
Tabel 7.5 Kasus Uji Validasi Mengirimkan Aspirasi	119
Tabel 7.6 Kasus Uji Validasi Mengelola Aspirasi	119
Tabel 7.7 Kasus Uji Validasi Melihat Statistik Aspirasi	120
Tabel 7.8 Kasus Uji Validasi Memberikan Tanggapan	120
Tabel 7.9 Kasus Uji Validasi Memantau Aspirasi	121
Tabel 7.10 Kasus Uji Validasi Mengelola Akun	121
Tabel 7.11 Kasus Uji Validasi Menghapus Aspirasi	122
Tabel 7.12 Hasil Uji Kompatibilitas	124
Tabel 7.13 Skenario UAT pada Mahasiswa/Lembaga	125
Tabel 7.14 Skenario UAT pada Operator	125
Tabel 7.15 Skenario UAT pada Administrator	126
Tabel 7.16 Hasil Kuesioner Pengujian Penerimaan Pengguna	127
Tabel 7.17 Bobot Nilai Pilihan Jawaban UAT	129
Tabel 7.18 Hasil Kuesioner UAT Setelah Diolah	129
Tabel 7.19 Hasil Analisa Kuesioner UAT	131
Tabel 7.20 Hasil perhitungan waktu yang dibutuhkan mahasiswa dan lembaga	134
Tabel 7.21 Hasil perhitungan waktu yang dibutuhkan anggota komisi III	135
Tabel 7.22 Persentase efisiensi waktu mahasiswa atau lembaga	136
Tabel 7.23 Persentase efisiensi waktu anggota komisi III DPM UB	137

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi DPM UB	8
Gambar 2.2 Diagram <i>Reuse-oriented software engineering</i>	12
Gambar 2.3 Piramida Persyaratan	18
Gambar 2.4 Hierarki Diagram UML.....	20
Gambar 2.5 Diagram <i>use case</i> utama	21
Gambar 2.6 Diagram <i>use case</i> pendukung	21
Gambar 2.7 Contoh <i>Sequence Diagram</i>	23
Gambar 2.8 Contoh <i>Class Diagram</i>	24
Gambar 2.9 Konsep <i>Model-view-controller</i>	25
Gambar 2.10 Jenis-jenis pengujian kompatibilitas	27
Gambar 2.11 Tahapan UAT	28
Gambar 2.12 UAT <i>Feedback Report Template</i>	29
Gambar 3.1 Diagram Metodologi Penelitian	30
Gambar 4.1 Proses bisnis pelayanan aspirasi melalui sekretariat DPM UB (<i>as-is</i>)	35
Gambar 4.2 Proses bisnis pelayanan aspirasi melalui DPM UB Safari Fakultas (<i>as-is</i>)	37
Gambar 4.3 Proses bisnis pelayanan aspirasi usulan (<i>to-be</i>).....	39
Gambar 4.4 Diagram <i>Use Case</i> SIAP Online DPM UB.....	61
Gambar 5.1 Rancangan Arsitektur Sistem	78
Gambar 5.2 Diagram Interaksi Login.....	79
Gambar 5.3 Diagram Interaksi Mengirimkan Aspirasi	80
Gambar 5.4 Diagram Interaksi Memantau Aspirasi.....	81
Gambar 5.5 Diagram Interaksi Memberikan Tanggapan.....	82
Gambar 5.6 Diagram Interaksi Mengelola Akun.....	83
Gambar 5.7 Diagram Kelas Model	84
Gambar 5.8 Diagram Kelas <i>Controller</i>	85
Gambar 5.9 Diagram Relasi Kelas <i>Model</i> dengan <i>Controller</i>	86
Gambar 5.10 Perancangan Basis Data	87
Gambar 5.11 Rancangan Halaman Utama	95
Gambar 5.12 Rancangan Halaman Profil	96

Gambar 5.13 Rancangan Halaman Kirim Aspirasi.....	96
Gambar 5.14 Rancangan Halaman Statistik Aspirasi	97
Gambar 5.15 Rancangan Halaman Kelola Data Aspirasi.....	98
Gambar 5.16 Rancangan Halaman Detail Aspirasi.....	98
Gambar 5.17 Rancangan Halaman Kelola Akun.....	99
Gambar 6.1 Antarmuka Halaman Utama.....	111
Gambar 6.2 Antarmuka Login Admin.....	111
Gambar 6.3 Antarmuka Profil	112
Gambar 6.4 Antarmuka Kirim Aspirasi.....	112
Gambar 6.5 Antarmuka Statistik Aspirasi	113
Gambar 6.6 Antarmuka Kelola Data Aspirasi.....	113
Gambar 6.7 Antarmuka Detail Aspirasi.....	114
Gambar 6.8 Antarmuka Cetak Aspirasi	114
Gambar 6.9 Antarmuka Kelola Akun.....	115
Gambar 6.10 Antarmuka Kelola Akun – Tambah Akun	115
Gambar 7.1 Hasil Pengujian Kompatibilitas.....	123
Gambar 7.2 Grafik Persentase Hasil Uji Kompatibilitas.....	124
Gambar 7.3 Grafik Persentase Hasil Uji Penerimaan Pengguna.....	132
Gambar 7.4 Grafik Efisiensi Waktu Mahasiswa atau Lembaga	136
Gambar 7.5 Grafik Efisiensi Waktu Anggota Komisi III DPM UB.....	137
Gambar 7.6 Grafik Rata-rata hasil uji Efisiensi Waktu	138

DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	143
LAMPIRAN B VERIFIKASI DAN VALIDASI PERSYARATAN FUNGSIONAL.....	145
LAMPIRAN C KUESIONER PENGUJIAN PENERIMAAN PENGGUNA	146
C.1 Kuesioner untuk Mahasiswa dan Lembaga	146
C.2 Kuesioner untuk Operator Sistem	147
C.3 Kuesioner untuk Administrator Sistem	148



BAB 1 PENDAHULUAN

1.1 Latar belakang

Dewan Perwakilan Mahasiswa Universitas Brawijaya atau disingkat DPM UB merupakan lembaga tinggi yang berada dalam struktur Lembaga Kedaulatan Mahasiswa Universitas Brawijaya (LKM UB) yang mewakili mahasiswa sebagai lembaga legislatif. DPM menjalankan tugas dan fungsi legislatif sebagaimana yang tertuang dalam anggaran dasar dan anggaran rumah tangga LKM UB. Fungsi pokok DPM UB adalah sebagai pengontrol, legislasi, advokasi dan penyelenggara Pemilu Raya LKM UB. Dalam menjalankan fungsinya DPM UB tak akan lepas dari mahasiswa, terutama dalam hal menyangkut aspirasi mahasiswa. Aspirasi sangat penting bagi DPM UB, karena dengan adanya aspirasi organisasi tersebut mendapatkan informasi-informasi yang berupa aduan, kritik, keluhan, saran dan lain-lain. Informasi tersebut digunakan untuk melakukan advokasi dalam merumuskan kebijakan atau pengambilan keputusan. Dalam melakukan advokasi, DPM UB menyediakan layanan bagi mahasiswa untuk menyampaikan aspirasinya secara langsung dengan cara mendatangi sekretariat DPM UB dan menemui komisi III selaku bagian advokasi. Anggota DPM UB juga melakukan safari fakultas yaitu dengan cara keliling fakultas untuk menyerap aspirasi yang dirasakan mahasiswa dan terkadang juga melakukan penyebaran kuesioner untuk mendapatkan tanggapan dari mahasiswa ketika terdapat permasalahan detail yang membutuhkan penanganan segera.

Berdasarkan hasil wawancara dengan salah satu anggota komisi III DPM UB, mengatakan bahwa terdapat kendala yang dihadapi, dalam satu bulan terakhir (maret 2018) ada dua belas kali aspirasi yang masuk dan tak ada satupun yang melewati jalur sekretariat DPM UB dengan alasan karena mahasiswa sungkan menyampaikan aspirasinya, mahasiswa masih terkendala harus datang ke ruang sekretariat dan bertatap muka langsung dengan anggota DPM UB. Proses ini dirasa rumit bagi mahasiswa sehingga mereka lebih memilih menunggu anggota DPM terjun langsung menanyakan permasalahannya. Akibatnya, aspirasi yang masuk menjadi sangat kurang dan berdampak pada kurang optimalnya kinerja DPM UB. Padahal menurut Siregar, et al. (2016) dalam artikelnya yang berjudul *"Implementasi Good Governance pada Organisasi Mahasiswa di Universitas Gadjah Mada Demi Mewujudkan Good Student Governance"* menyatakan bahwa peran aktif mahasiswa dalam berpartisipasi menjalankan maupun mengawasi organisasi mahasiswa sangat dibutuhkan untuk mengoptimalkan kinerja organisasi tersebut demi terwujudnya fungsi dan tujuan organisasi.

Dari permasalahan diatas dapat diidentifikasi bahwa dua akar utama penyebab kurangnya aspirasi mahasiswa adalah: Pertama, banyaknya mahasiswa yang memang apatis/kurang peduli terhadap lembaga kemahasiswaan yang ada, karena berkaca pada artikelnya Siregar, et al. (2016) menunjukkan bahwa data jumlah partisipasi mahasiswa dalam mengawasi organisasi mahasiswa persentasenya untuk organisasi tingkat fakultas sebanyak 25% sedangkan untuk

tingkat universitas hanya 7% dengan kata lain partisipasinya masih sangat rendah. Sedangkan penyebab yang kedua, karena pelayanan yang diberikan oleh DPM UB kurang maksimal sehingga berdampak pada lambatnya dalam menyerap aspirasi mahasiswa, dapat dilihat dari perbandingan jumlah aspirasi yang masuk dalam satu bulan terakhir dengan jumlah anggota DPM UB yang ditugaskan menangani aspirasi sebanyak lima orang belum termasuk staff ahlinya, hasil ini menunjukkan masih sangat kurang. Anggota DPM UB beralasan tidak mampu menemui semua elemen-elemen mahasiswa secara langsung karena disibukkan dengan agenda-agenda lain termasuk karena statusnya juga masih sebagai mahasiswa aktif yang disibukkan dengan tugas dan jadwal mata kuliah, padahal kebutuhan aspirasi tidak hanya didominasi oleh mahasiswa tertentu saja, semua mahasiswa berhak menyampaikan aspirasinya. Maka dari itu diperlukanlah tata kelola organisasi mahasiswa yang baik, yang dapat melayani kebutuhan aspirasi mahasiswa dengan mudah dan mampu mengurangi sikap ketidakpedulian mahasiswa terhadap kinerja suatu organisasi mahasiswa. Untuk menerapkan *Good Student Governance* idealnya membutuhkan sebuah instrumen yang dapat dipakai yaitu melalui pemanfaatan media dan teknologi informasi (Siregar, et al., 2016).

Berdasarkan observasi berupa wawancara yang dilakukan anggota DPM UB, mahasiswa membutuhkan adanya suatu wadah yang memungkinkan dapat mengakomodasi penerimaan aduan atau aspirasi dari permasalahan yang sedang mereka rasakan dan mahasiswa juga dapat memantau perkembangan penyelesaiannya melalui akses dimana saja dan kapan saja secara *online*. Sedangkan bagi DPM UB karena saat ini belum adanya dukungan teknologi yang dapat membantu mereka maka dibutuhkan sistem informasi yang dapat menyerap aspirasi mahasiswa, efektif dan efisien dalam proses maupun pendokumentasiannya. Harapannya dapat menambah minat mahasiswa untuk peduli terhadap organisasi dan dapat mengoptimalkan kinerja DPM UB sebagai lembaga legislatif. Menurut Prihantoro, et al. (2011) dalam artikelnya, adanya sistem informasi dalam sebuah lembaga organisasi kemahasiswaan akan lebih mantap dalam mengakarkan perannya sebagai corong informasi dan advokasi publik mahasiswa, serta berbagai program kerja yang akan dilaksanakan menjadi lebih efektif, mudah, dan tentunya unggul dalam bidang *Information and communication technology (ICT)*.

Dari uraian permasalahan tersebut, maka dibutuhkanlah sebuah sistem informasi yang dapat menunjang kebutuhan mahasiswa ataupun kebutuhan dari DPM UB itu sendiri. Oleh karena itu dilakukan penelitian ini, dengan harapan dapat mempermudah dan menambah minat mahasiswa dalam menyampaikan aspirasinya sehingga dapat mengoptimalkan kinerja dari lembaga tersebut. Sistem yang akan dikembangkan ini, memiliki kemiripan fungsi dengan sistem "LAPOR BEM FILKOM" yang juga sama digunakan untuk menyerap aspirasi dari mahasiswa namun ruang lingkupnya sangatlah berbeda. Berdasarkan adanya kemiripan tersebut, maka akan dilakukan analisis untuk mendapatkan komponen-komponen yang dapat digunakan ulang agar dapat mempercepat dalam proses pengembangannya. Oleh karena periode DPM UB yang hanya satu tahun, minimnya dana yang dimiliki dan dibutuhkan sistem yang berkualitas aman dari

ancaman peretas yang bisa mengganggu dalam pelayanannya, maka dibutuhkan suatu metode yang dapat menghemat waktu dalam pengembangan dan kualitas sistemnya tetap terjaga.

Berdasarkan kebutuhan tersebut, penelitian ini bertujuan untuk mengembangkan sistem menggunakan pemodelan *reuse-oriented development*, karena dalam proses pengembangannya berfokus untuk menggunakan komponen-komponen yang dapat digunakan ulang daripada mengembangkannya dari nol. Pendekatan ini didasarkan pada adanya sejumlah komponen *reusable* yang sangat signifikan. Menurut Baldassarre, et al. (2005) semua pengembangan berorientasi penggunaan ulang mengarah ke kualitas dan produktivitas yang lebih tinggi dibandingkan dengan metode yang masih konvensional (*waterfall*). Rekayasa perangkat lunak berorientasi penggunaan ulang memiliki keuntungan nyata untuk mengurangi jumlah fungsi yang akan dikembangkan, mengurangi risiko, biaya dan pengembangannya akan lebih cepat (Sommerville, 2011). Oleh karena itu, pemodelan ini perlu digunakan agar menghemat waktu dan tenaga kerja dalam proses pengembangannya.

1.2 Rumusan masalah

Berdasarkan uraian pada latar belakang, rumusan masalahnya secara umum adalah dapatkah dikembangkan suatu sistem informasi yang mampu membantu pelaksanaan proses-proses bisnis di dalam menangani pelayanan aspirasi mahasiswa UB, khususnya dalam proses penyampaian dan penyerapannya. Masalah ini diharapkan dapat terselesaikan dengan bantuan sejumlah fungsi yang ditawarkan oleh Sistem Informasi Aspirasi *Online* (SIAP *Online*) DPM UB. Untuk menjawab rumusan masalah umum tersebut maka perlu untuk menjawab beberapa pertanyaan penelitian berikut ini:

1. Bagaimanakah hasil pemodelan proses bisnis pelayanan aspirasi di DPM UB?
2. Bagaimanakah hasil analisis persyaratan SIAP *Online* DPM UB yang sesuai dengan kebutuhan organisasi tersebut?
3. Bagaimanakah hasil rancangan dan implementasi SIAP *Online* DPM UB menggunakan pemodelan *reuse-oriented development* yang sesuai dengan spesifikasi persyaratan sistem tersebut?
4. Bagaimanakah hasil uji SIAP *Online* DPM UB terkait persyaratan fungsionalitas, non fungsionalitas dan tingkat penerimaan pengguna dalam melayani aspirasi mahasiswa?

1.3 Tujuan

Tujuan umum penelitian ini adalah untuk mengembangkan SIAP *Online* DPM UB agar dapat membantu pelayanan aspirasi mahasiswa. Sedangkan tujuan khusus dalam penelitian ini adalah sebagai berikut:

1. Memodelkan proses bisnis pelayanan aspirasi di DPM UB.
2. Menganalisis dan menyusun spesifikasi persyaratan SIAP *Online* DPM UB.

3. Merancang dan mengimplementasikan rancangan SIAP *Online* DPM UB menggunakan pemodelan *reuse-oriented development*.
4. Menguji terkait persyaratan fungsionalitas, non fungsionalitas dan tingkat penerimaan pengguna terhadap SIAP *Online* DPM UB.

1.4 Manfaat

Adapun beberapa manfaat yang diharapkan dengan adanya hasil penelitian ini, antara lain:

1. Bagi DPM UB:
 - a. Membantu mewujudkan sistem informasi aspirasi mahasiswa.
 - b. Membantu menyerap dan mendokumentasikan aspirasi mahasiswa.
 - c. Membantu mewujudkan Visi Misi DPM UB.
 - d. Menjadikan DPM yang unggul dibidang *Information and communication technology (ICT)*.
2. Bagi mahasiswa dan lembaga sebagai pengguna sistem:
 - a. Mendapatkan media baru dalam menyampaikan aspirasinya ke DPM UB.
 - b. Memudahkan dan mempercepat dalam menyampaikan aspirasinya.

1.5 Batasan masalah

Berdasarkan rumusan masalah, maka terdapat batasan-batasan masalah yang digunakan dalam penelitian ini, yaitu:

1. Penelitian dilakukan hanya pada pelayanan aspirasi studi kasus pada DPM UB.
2. Responden yang digunakan adalah Mahasiswa aktif Universitas Brawijaya baik yang berperan aktif dalam organisasi maupun tidak.
3. Metode pengembangan sistem menggunakan pemodelan *Reuse-Oriented Development*.

1.6 Sistematika pembahasan

Sistematika pembahasan dalam penyusunan laporan ditunjukkan untuk menggambarkan uraian dari penelitian ini, yang secara garis besar meliputi bab-bab berikut ini:

BAB I PENDAHULUAN

Memuat latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

BAB II LANDASAN KEPUSTAKAAN

Menguraikan tentang kajian pustaka dan dasar teori yang mendasari Pengembangan Sistem Informasi Aspirasi *Online* Berbasis *Web* menggunakan pemodelan *Reuse-Oriented Development* (Studi Kasus: DPM Universitas Brawijaya).

BAB III METODOLOGI PENELITIAN

Membahas mengenai metode yang digunakan dalam penelitian yang terdiri dari studi literatur, pemodelan proses bisnis, analisis persyaratan, analisis komponen, modifikasi persyaratan, perancangan dengan komponen, implementasi dan integrasi, pengujian dan analisis, serta penulisan laporan.

BAB IV PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN

Membahas pemodelan proses bisnis dari proses bisnis *as-is* hingga ke proses bisnis *to-be*, membahas analisis persyaratan yang terdiri dari elisitasi persyaratan, spesifikasi persyaratan, verifikasi dan validasi. Dalam bab ini juga membahas pemodelan *use case* dari tahap identifikasi aktor hingga spesifikasi *use case*.

BAB V ANALISIS KOMPONEN DAN PERANCANGAN

Membahas mengenai analisis komponen, modifikasi persyaratan terhadap komponen yang digunakan, serta membahas perancangan sistem dengan menggunakan komponen *reuse*.

BAB VI IMPLEMENTASI

Membahas implementasi kode program berdasarkan perancangan dan mengintegrasikan komponen *reuse* untuk membentuk sistem yang baru.

BAB VII PENGUJIAN DAN ANALISIS

Membahas proses pengujian sistem dan jawaban atas pertanyaan yang diuraikan dalam sub-bab rumusan masalah. Bab ini juga menjelaskan hasil analisis dari validasi sistem dengan kesesuaian kebutuhan organisasi serta tinjauan pengembangan sistem.

BAB VIII PENUTUP

Memuat ringkasan hasil dalam bentuk kesimpulan yang diperoleh dari pembuatan dan saran-saran untuk pengembangan selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Berdasarkan pada judul penelitian ini, mengenai *“Pengembangan Sistem Informasi Aspirasi Online Berbasis Web Menggunakan Pemodelan Reuse-oriented Development (Studi kasus: DPM Universitas Brawijaya)”* maka diperlukan pengetahuan dari penelitian sebelumnya untuk mengetahui urgensi dan pemodelan fungsi dari sistem yang akan dikembangkan. Penulis dalam hal ini mencoba mempelajari artikel Siregar, et al. (2016) yang berjudul *“Implementasi Good Governance pada Organisasi Mahasiswa di Universitas Gadjah Mada Demi Mewujudkan Good Student Governance”* menyatakan bahwa peran aktif mahasiswa dalam berpartisipasi menjalankan maupun mengawasi organisasi mahasiswa sangat dibutuhkan untuk mengoptimalkan kinerja organisasi tersebut demi terwujudnya fungsi dan tujuan organisasi. Dia juga menyatakan bahwa untuk menerapkan *Good Student Governance* idealnya membutuhkan sebuah instrumen yang dapat dipakai yakni melalui pemanfaatan media dan teknologi informasi.

Selanjutnya penulis juga perlu mencari perbandingan referensi metode yang cocok untuk proses pengembangannya. Baldassarre, et al. (2005) dalam artikelnya yang berjudul *“An Industrial Case Study on Reuse Oriented Development”* menyatakan bahwa semua pengembangan berorientasi penggunaan ulang yang ditelitinya mengarah ke kualitas dan produktivitas yang lebih tinggi dibandingkan dengan metode yang masih konvensional (*waterfall*) namun dengan catatan adanya komponen *reusable*. Dalam artikel lain yang berjudul *“Framework for Better Reusability in Component Based Software Engineering”* menyatakan bahwa untuk menghindari resiko sistem dalam pengembangan perangkat lunak yang lebih kompleks dapat dibangun dari komponen yang lebih kecil. Komponen-komponen yang lebih kecil ini diintegrasikan untuk membuat keseluruhan sistem (Tahir, et al., 2016).

2.2 Gambaran Organisasi

2.2.1 Profil Organisasi

Organisasi ini bernama Dewan Perwakilan Mahasiswa Universitas Brawijaya atau disingkat DPM UB yang merupakan lembaga tinggi legislatif yang terdapat dalam kehidupan kemahasiswaan di Universitas Brawijaya. Lembaga ini merupakan bagian dari alat kelengkapan Lembaga Kedaulatan Mahasiswa Universitas Brawijaya (LKM UB). Didirikan di Malang pada tanggal 27 September 1998, sampai waktu yang tidak ditentukan. Bertempat dan berkedudukan di lingkungan kampus Universitas Brawijaya Malang. Kedaulatan tertinggi LKM UB berada di tangan mahasiswa Universitas Brawijaya yang berasaskan Pancasila. LKM UB berprinsip pada kedaulatan mahasiswa, menjunjung tinggi intelektual, pemberdayaan mahasiswa, rekonstruksi lembaga secara mendasar ke arah yang lebih baik dan otonomi kelembagaan.

2.2.2 Visi Organisasi

Visi Dewan Perwakilan Mahasiswa Universitas Brawijaya adalah “Terwujudnya DPM UB yang konstruktif, progresif dan interaktif”.

2.2.3 Misi Organisasi

Misi Dewan Perwakilan Mahasiswa Universitas Brawijaya:

1. Menjalankan manajemen administrasi internal DPM UB secara berkelanjutan dan profesional.
2. Membangun komunikasi aktif dan kedekatan dengan konstituen serta eksternal DPM UB.
3. Aktif dalam mengawal kebijakan dan kinerja lembaga eksekutif mahasiswa.
4. Meningkatkan peran lembaga legislatif melalui produk hukum DPM UB yang efektif.
5. Menjaring dan mengawal aspirasi mahasiswa secara aktif dan tanggap.

2.2.4 Tugas dan Wewenang

Adapun tugas dan wewenang Dewan Perwakilan Mahasiswa Universitas Brawijaya adalah sebagai berikut:

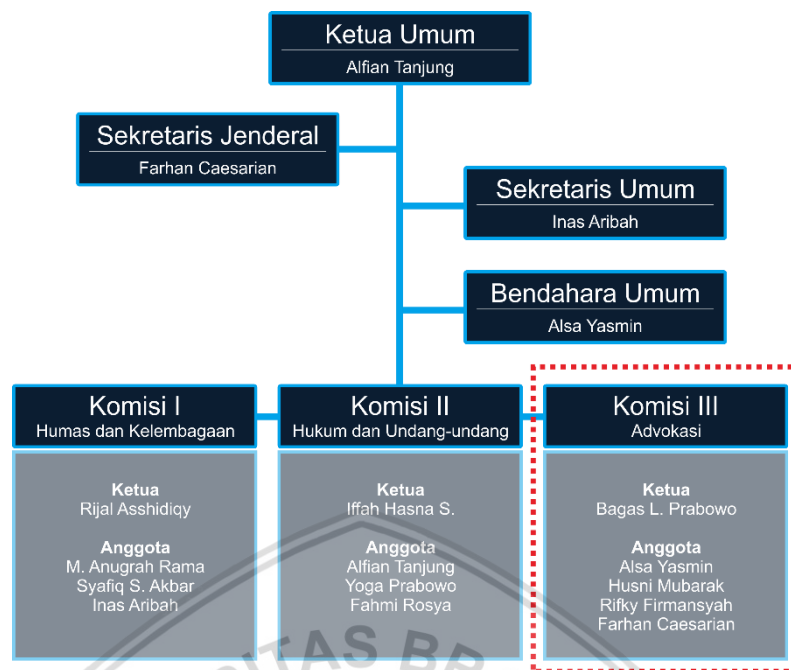
1. Mengawasi EM UB dalam melaksanakan GBHK LKM UB, ketetapan KM UB, Undang-undang DPM UB dan peraturan LKM UB lainnya.
2. Menyerap dan merumuskan aspirasi anggota LKM UB dan menyalurkan kepada EM UB serta melakukan advokasi kepada pihak yang terkait.
3. Bila dalam pandangan DPM UB, EM UB tidak melaksanakan tugasnya atau menyimpang dari arah kebijakan KM UB, maka DPM UB berwenang mengajukan memorandum I dengan batas waktu tiga puluh hari. Kemudian jika EM UB tetap melakukan penyimpangan, maka DPM UB berwenang mengajukan memorandum II dengan batas waktu lima belas hari, setelah batas waktu tersebut EM UB tidak memperbaiki, maka dapat dilaksanakan Sidang Istimewa oleh KM UB.
4. Mengajukan rancangan GBHK kepada KM UB.
5. Membuat dan menetapkan Undang-Undang untuk melaksanakan tugas sebagai lembaga tinggi legislatif.
6. Membuat dan menetapkan peraturan di bawah Undang-Undang.
7. Bertanggung jawab dalam penyelenggaraan Pemilu Raya LKM UB.

2.2.5 Lokasi Organisasi

Organisasi ini bertempat dan berkedudukan di dalam kampus Universitas Brawijaya Malang.

2.2.6 Struktur Organisasi

Keanggotaan DPM UB adalah para kandidat yang terpilih pada Pemilu Raya. Anggota-anggota DPM UB dibagi menjadi komisi-komisi dengan masa jabatan satu tahun dan sesudahnya dapat dipilih kembali sebanyak-banyaknya satu periode lagi melalui Pemilu Raya.



Gambar 2.1 Struktur Organisasi DPM UB

Sumber: DPM UB (2018)

Gambar 2.1 menjelaskan struktur organisasi di DPM UB periode saat ini yang diketuai oleh Alfian Tanjung. Berdasarkan gambar tersebut, terdapat tiga komisi yang memiliki fungsi masing-masing.

Komisi yang terlibat langsung dalam pengembangan sistem pada penelitian ini adalah komisi III yang memiliki fungsi advokasi. Tugas dari komisi III adalah menyerap dan juga merumuskan aspirasi dari anggota Lembaga Kedaulatan Mahasiswa (LKM) UB lalu menyalurkan kepada Eksekutif Mahasiswa (EM) UB serta mengadvokasi kepada pihak-pihak terkait. Data dari hasil penyerapan aspirasi mahasiswa yang dilakukan oleh komisi III juga digunakan oleh komisi lain dalam merumuskan kebijakan atau pengambilan keputusan. Saat ini, komisi III DPM UB diketuai oleh Bagas L. Prabowo dan anggotanya sebanyak empat orang yaitu: Alsa Yasmin, Husni Mubarak, Rifky Firmansyah dan Farhan Caesarian.

2.3 Definisi Aspirasi *Online*

2.3.1 Aspirasi

Aspirasi yang berasal dari kata *aspire*, artinya menginginkan atau bercita-cita. Dalam kamus besar bahasa Indonesia (KBBI) menyebutkan bahwa aspirasi adalah harapan dan tujuan untuk keberhasilan pada masa yang akan datang (KBBI, 2018).

Aspirasi merupakan harapan dan keinginan individu akan suatu keberhasilan atau suatu prestasi. Arti aspirasi adalah suatu keinginan sangat kuat yang ditandai dengan usaha untuk meraih sesuatu hal yang dipandang lebih bernilai dan lebih tinggi dari keadaan sekarang. Aspirasi mengarahkan aktivitas individu untuk menjadi lebih fokus akan pencapaian tujuan-tujuan tersebut. Keinginan dalam

aspirasi bisa berupa peningkatan status individu maupun keinginan yang sifatnya ekstrim, terlalu berani ataupun tidak wajar (Hurlock, 1979).

Pengertian aspirasi dalam konteks mahasiswa adalah berbagai tuntutan untuk mengusulkan suatu proses perubahan atas sesuatu hal yang dikemas ke dalam ide kreatif mahasiswa. Pembentukan aspirasi pada umumnya dibuat oleh senat atau lembaga-lembaga yang mewakili mahasiswa.

2.3.1.1 Jenis-jenis aspirasi

Terdapat dua jenis aspirasi menurut Hurlock (1979) yang dikemukakan berdasarkan sifatnya, yaitu:

- a. Aspirasi positif yaitu keinginan dalam meraih kemampuan. Orang yang menyampaikan aspirasi positif merupakan orang yang ingin lebih baik atau lebih tinggi dari pada keadaannya yang sekarang.
- b. Aspirasi negatif yaitu keinginan untuk mempertahankan apa yang sudah dicapai saat ini, tanpa meningkatkan keinginannya.

Aspirasi berdasarkan tujuannya dibedakan menjadi dua macam (Hurlock, 1979), yaitu:

- a. *Immediate aspiration* (langsung) merupakan aspirasi yang cita-cita/tujuan yang ingin dicapai seseorang dalam waktu yang dekat atau tidak terlalu lama (sekarang, besok, minggu depan, atau bulan depan).
- b. *Remote corporation* (jauh) merupakan aspirasi yang memiliki tujuan jauh atau ingin tercapai di masa mendatang.

2.3.2 Online

Online dalam bahasa Indonesia adalah dalam jaringan yang disingkat menjadi daring, *online* memiliki arti tertentu dalam hal telekomunikasi dan teknologi komputer. Namun, secara umum menunjukkan keadaan seseorang yang terhubung. *Online* dapat juga diartikan sebagai keadaan suatu komputer yang sudah saling terhubung sehingga dapat saling bertukar informasi.

Berdasarkan dari penjelasan diatas kata aspirasi *online* dalam penelitian ini dapat disimpulkan bahwa permintaan oleh pihak berkepentingan kepada pejabat berwenang dalam kasus ini adalah DPM UB, untuk menindak menurut undang-undang yang digunakan pada LKM UB terhadap seseorang atau lembaga yang telah melakukan tindakan merugikan atau ketidakpuasan terhadap layanan yang diberikan secara terhubung melalui teknologi dan telekomunikasi.

2.4 Customer Relationship Management (CRM)

CRM adalah suatu proses tatakelola interaksi antara perusahaan dengan pelanggan atau dengan calon pelanggan, yang dalam kasus ini merupakan LKM UB dengan mahasiswa. Pengelolaan interaksi perusahaan terhadap pelanggan termasuk suatu sistem yang dikhususkan untuk memberikan layanan informasi kepada pelanggan yang berupa *call center* ataupun dukungan teknis lainnya. CRM

merupakan suatu pendekatan terintegrasi yang bertujuan untuk mengidentifikasi, memperoleh dan mempertahankan pelanggan (Buttle, 2009). CRM memiliki tujuan utama untuk meningkatkan atau memaksimalkan keuntungan dan pertumbuhan perusahaan dalam jangka panjang dengan memahami secara lebih baik atas perilaku atau keinginan pelanggan untuk menciptakan umpan balik pelanggan yang lebih efektif bagi perusahaan.

2.4.1 Jenis-Jenis CRM

Berikut ini merupakan jenis *customer relationship management* (CRM) menurut Buttle (2009):

- a. *Strategic*, bertujuan agar memaksimalkan strategi pengembangan bisnis yang berfokus kepada pelanggan. Strategi ini dilakukan dengan cara menciptakan dan memberikan nilai yang lebih baik dari pesaing, untuk mendapatkan sekaligus mempertahankan pelanggan.
- b. *Operational*, difokuskan untuk mengotomatisasi dan mengembangkan layanan ke dalam bentuk aplikasi perangkat lunak. Tujuan dari aplikasi ini untuk melayani kebutuhan pelanggan seperti pemasaran, penjualan dan layanan pelanggan dapat terintegrasi dan dilakukan secara otomatis.
- c. *Analytical*, berkaitan dengan proses penyimpanan, ekstraksi, integrasi, proses, distribusi dan pembuatan laporan data mengenai pelanggan yang memiliki tujuan untuk meningkatkan daya saing dan laba perusahaan.
- d. *Collaborative*, penyalarsan antara strategi dengan taktis untuk mengoptimalkan kinerja perusahaan, mitra dan nilai pelanggan yang menggunakan teknologi sebagai perantara masing-masing organisasi di dalam suatu perusahaan.

Dalam hal ini DPM UB ingin menerapkan CRM *Operational* untuk melayani aspirasi dan aduan mahasiswa secara otomatis dilakukan oleh sistem. Sehingga dapat mengurangi beban kerja dari DPM UB itu sendiri.

2.4.2 CRM Operational

CRM *operational* merupakan pengembangan dan otomatisasi suatu layanan yang bertujuan untuk membantu organisasi atau perusahaan dalam memberikan layanan pelanggan secara otomatis dan terintegrasi dengan menggunakan aplikasi. Dalam kasus ini DPM UB menerapkan CRM *operational* untuk mengotomatisasi kebutuhan mahasiswa dalam menyampaikan aspirasi menggunakan sistem informasi yang berbasis *web*. Dengan CRM *operational* mahasiswa atau lembaga tidak perlu bertemu lagi secara langsung dengan anggota DPM UB, khususnya dalam proses penyampaian dan penyerapan aspirasi.

2.4.3 Complaint Management

Complaint management dalam kasus ini merupakan bagian dari CRM *operational* yang memiliki tujuan untuk menerima dan mengelola aspirasi atau aduan mahasiswa/lembaga yang berkaitan dengan layanan LKM UB. Dengan

adanya manajemen keluhan (aspirasi) mahasiswa, DPM UB memiliki kesempatan untuk menyiapkan diri dalam menerima berbagai macam keluhan mahasiswa dan menyediakan layanan penanganan untuk menjaga kualitas pelayanan yang diberikan. Suatu organisasi harus mencatat segala informasi pelanggan (mahasiswa) dan menyimpannya ke dalam format yang terstruktur, isi informasi tersebut dapat berupa rincian identitas pengeluh/pelapor, alasan menyampaikan keluhan, tujuan keluhan dan tindakan yang dapat diambil. Melalui rincian informasi tersebut DPM UB dapat menjadikan dasar bagi organisasi dalam mengambil keputusan, tindakan ataupun kebijakan yang diperlukan selanjutnya. Untuk menerapkan *electronic customer relationship management* (e-CRM) dalam CRM *operational* dapat dengan menerapkan suatu sistem informasi untuk menyimpan, mengelola dan menjawab aspirasi atau aduan dari mahasiswa.

2.5 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (RPL) merupakan penggunaan prinsip-prinsip teknik keahlian dalam pembuatan suatu perangkat lunak untuk mendapatkan perangkat lunak yang ekonomis, handal dan bekerja secara efisien pada mesin yang sebenarnya (Pressman, 2010). Menurut Sommerville (2011) RPL merupakan suatu prinsip perekayasaan yang memiliki hubungan dengan aspek pembuatan perangkat lunak, dari tahapan awal spesifikasi sistem hingga perawatan sistem setelah digunakan.

Suatu proses merupakan dasar dalam perekayasaan perangkat lunak. Proses dalam RPL mendefinisikan *framework* (kerangka kerja) dalam manajemen kontrol dan menetapkan metode teknis yang akan digunakan. Metode RPL dapat meliputi analisis persyaratan, desain, konstruksi program, pengujian dan dukungan. *Tools* RPL dapat memberikan dukungan otomatis atau semi otomatis untuk proses dan metode pengembangan.

2.6 Software Process Model

Software process model merupakan pendekatan sistematis yang dapat digunakan dalam pengembangan perangkat lunak dan merupakan suatu aktivitas terurut menuju produksi dari pembuatan produk perangkat lunak (Sommerville, 2011). *Software process model* adalah suatu representasi model proses pengembangan perangkat lunak yang dipresentasikan, disederhanakan, dan perspektif secara khusus. Misalnya, model aktivitas proses menunjukkan aktivitas dan urutannya namun mungkin tidak menunjukkan peran orang-orang yang terlibat di dalam aktivitas ini.

Fungsi utama dari pemodelan proses pengembangan perangkat lunak sebagai berikut:

- a. Menentukan tahapan yang perlu dilakukan dalam pengembangan perangkat lunak.
- b. Menentukan urutan tahapan pelaksanaan dalam kerangka pengembangan perangkat lunak.

- c. Menentukan perpindahan atau kriteria transisi dari suatu tahapan ke tahapan berikutnya.

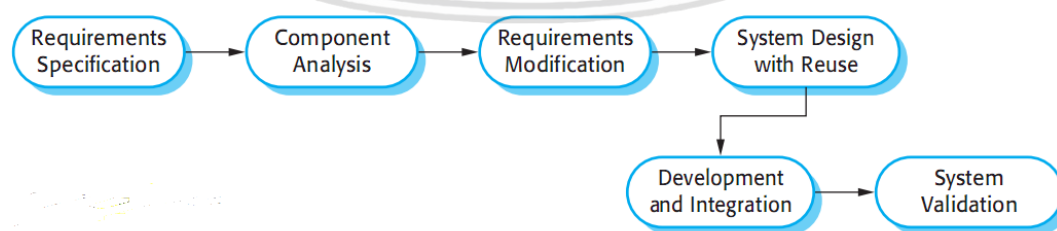
Sommerville (2011) dalam bukunya membahas tiga model yang tidak saling eksklusif dan sering digunakan bersamaan, terutama untuk pengembangan sistem yang besar, yaitu: *waterfall model*, *incremental development* dan *reuse-oriented software engineering*. Namun pada penelitian kali ini penulis hanya fokus pada model *reuse-oriented* dikarenakan dalam pengembangannya akan menggunakan pemodelan ini.

2.6.1 Reuse-Oriented Development

Reuse-oriented development atau *reuse-oriented software engineering* menurut Sommerville (2011) adalah pendekatan yang didasarkan pada adanya sejumlah komponen *reusable* yang signifikan. Proses pengembangan ini berfokus untuk mengintegrasikan komponen-komponen yang ada ke dalam sistem daripada mengembangkannya dari nol.

Sebagian besar proyek pengembangan perangkat lunak, menggunakan komponen penggunaan ulang dari perangkat lunak lain. Ini sering terjadi secara informal ketika pengembang proyek memiliki kode atau mengetahui desain serupa dengan yang dibutuhkan. Mereka mencari komponen yang sama, memodifikasinya sesuai persyaratan dan memasukkannya ke dalam sistem baru mereka (Sommerville, 2011).

Penggunaan ulang ini dapat dilakukan terlepas dari model proses pengembangan yang digunakan. Namun di abad ke-21, pengembangan perangkat lunak yang berfokus pada pengembangan penggunaan ulang menjadi sangat sering digunakan (Sommerville, 2011). Pendekatan perangkat lunak berorientasi penggunaan ulang bergantung pada basis komponen yang dapat digunakan ulang dan kerangka kerja gabungan untuk komposisi komponennya. Terkadang, komponen yang digunakan adalah sistem dengan hak mereka sendiri (COTS atau *commercial off-the-shelf systems*) yang mungkin menyediakan fungsi khusus seperti *spreadsheet* atau pengolah kata.



Gambar 2.2 Diagram *Reuse-oriented software engineering*

Sumber: Sommerville (2011)

Pada Gambar 2.2 menunjukkan diagram alir model *Reuse-oriented development* yang terdiri dari tahapan: *Requirements specification*, *Component analysis*, *Requirements modification*, *System design with reuse*, *Development and Integration* dan *System validation*. Meskipun tahapan spesifikasi persyaratan awal

dan tahapan validasi sistem sebanding/sama dengan model lainnya, namun pada tahapan *intermediate* dalam proses *reuse-oriented*-nya sangatlah berbeda. Berikut penjelasan dari tahapan-tahapannya (Sommerville, 2011):

1. Analisis komponen (*Component analysis*). Berdasarkan spesifikasi persyaratan, analisis komponen dilakukan untuk mencari komponen yang sesuai dengan spesifikasi. Biasanya, tidak ada kecocokan yang tepat dan komponen yang dapat digunakan hanya menyediakan beberapa fungsi dari yang diperlukan.
2. Modifikasi persyaratan (*Requirements modification*). Pada tahapan ini, persyaratan dianalisis dengan menggunakan informasi mengenai kegunaan komponen yang telah ditemukan. Kemudian memodifikasi persyaratan untuk merefleksikan komponen yang tersedia. Jika persyaratan tidak mungkin untuk dimodifikasi, maka analisis komponen dapat dilakukan kembali untuk mencari solusi alternatif.
3. Perancangan sistem menggunakan *reuse* (*System design with reuse*). Pada tahapan ini, kerangka kerja sistem dirancang atau menggunakan ulang kerangka sistem yang telah ada. Perancang akan mempertimbangkan komponen yang dapat digunakan ulang dan memodifikasi kerangka kerja tersebut agar sesuai persyaratan. Apabila tidak tersedia komponen *reuse* beberapa kerangka baru memungkinkan untuk dirancang baru.
4. Pengembangan dan integrasi (*Development and Integration*). Ketika komponen tidak dapat diperoleh secara eksternal, maka diperlukan suatu pengembangan baru. Sedangkan komponen-komponen yang tersedia dari sistem COTS (*Commercial Off-The-Shelf system*) akan diintegrasikan agar dapat membentuk sistem yang baru. Dalam melakukan integrasi sistem di pemodelan ini, mungkin akan lebih baik menjadi bagian dari proses pengembangan dari pada menjadi bagian yang terpisah.

Commercial Off-The-Shelf system (COTS) merupakan suatu sistem perangkat lunak yang dapat disesuaikan dengan persyaratan pelanggan yang berbeda tanpa mengubah *source code* sistem. Sebagian besar perangkat lunak desktop dan berbagai produk lain servernya merupakan COTS, karena dirancang untuk digunakan umum maka biasanya mencakup banyak fitur dan fungsi. Oleh karena itu, memiliki potensi untuk digunakan ulang di lingkungan yang berbeda atau sebagai bagian dari aplikasi yang berbeda (Sommerville, 2011).

Reuse-oriented development adalah pendekatan pengembangan yang mencoba untuk memaksimalkan komponen yang ada. Unit perangkat lunak yang dapat digunakan ulang memiliki ukuran yang sangat berbeda. Sebagai contoh (Sommerville, 2011):

1. *Application system reuse*, seluruh sistem aplikasi dapat digunakan ulang dengan menggabungkannya tanpa mengubahnya ke sistem lain atau dengan mengkonfigurasi aplikasi untuk pelanggan berbeda.

2. *Component reuse* (komponen ulang dari sebuah aplikasi), mulai dari subsistem ke objek tunggal dapat digunakan ulang. Sebagai contoh, sistem pencocokan pola dikembangkan sebagai bagian dari sistem *text-processing* yang dapat digunakan ulang ke dalam sistem manajemen basis data.
3. *Object and function reuse*, komponen perangkat lunak yang mengimplementasikan fungsi tunggal, seperti fungsi matematika atau kelas objek juga dapat digunakan ulang. Bentuk penggunaan ulang ini, berdasarkan *standart libraries*. Banyak *libraries* dari fungsi dan kelas tersedia secara gratis. Pengembang dapat menggunakan ulang kelas dan fungsi ini dengan menautkannya ke kode pengembangan aplikasi yang baru. Seperti algoritma matematika dan grafik, dimana keahlian khusus diperlukan untuk mengembangkan objek dan fungsi yang efisien, maka dengan pendekatan ini akan menjadi sangat efektif.

Tiga jenis komponen *reuse* yang dapat digunakan ulang dalam pemodelan *reuse-oriented* (Sommerville, 2011):

1. *Web services*, yang dikembangkan sesuai dengan standar layanan dan yang tersedia untuk permintaan jarak jauh.
2. *Collections of object*, yang dikembangkan sebagai paket untuk diintegrasikan dengan kerangka komponen.
3. *Stand-alone software systems*, yang dikonfigurasi untuk digunakan dalam lingkungan tertentu.

Pengembangan perangkat lunak dari komponen yang ada dilakukan untuk mengurangi waktu dan biaya dari perangkat lunak. Modul dan kelas yang dapat digunakan ulang dapat mengurangi waktu implementasi, meningkatkan kemungkinan bahwa pengujian dan penggunaan sebelumnya telah menghilangkan bug dan melokalisasi modifikasi kode ketika perubahan dalam implementasi diperlukan (Tahir, et al., 2016).

Pemodelan *reuse-oriented development* merupakan pengembangan sistem dengan penggunaan ulang, informasi yang didapat dari pengembangan sistem sebelumnya menjadi bagian terpenting, dengan tujuan untuk meminimalisasi usaha yang diperlukan dalam pengembangan yang baru. Ide utama di balik pengembangan *reuse* adalah untuk membangun sistem dengan kualitas yang lebih tinggi dan biaya yang relatif rendah. Proses penggunaan ulang dapat mengurangi biaya dan waktu yang diperlukan dalam pengembangan sistem untuk sebuah organisasi skala besar jika dibandingkan dengan pengembangan secara independen. Tahapan *reuse* digunakan untuk membangun sistem baru berdasarkan aset yang telah ada. Sehingga, penggunaan ulang komponen sistem sebelumnya dapat meningkatkan dan membuat lebih baik serta akan menjawab kebutuhan user yang belum terpenuhi pada sistem sebelumnya.

Berikut keuntungan menggunakan *reuse-oriented development*:

1. Dapat mengurangi biaya pengembangan.
2. Dapat menghemat waktu pengembangan.

3. Apabila dikerjakan secara hati-hati, dapat meminimisasi kemungkinan kesalahan.
4. Kesalahan diketahui karena menggunakan komponen dari sistem sebelumnya.

Kekurangan menggunakan *reuse-oriented development*:

1. Tidak selalu praktis karena bisa saja terdapat komponen yang tidak tersedia, karena itu komponen baru harus dirancang.
2. Jika pengerjaan tidak serius dilakukan, dapat menghasilkan produk yang tidak sesuai dengan persyaratan yang diharapkan oleh pengguna.

2.7 Pemodelan Proses Bisnis

Pemodelan proses bisnis merupakan kumpulan aktivitas yang membutuhkan satu atau lebih masukan dan menghasilkan keluaran yang mempunyai nilai bagi pelanggan sekaligus dapat mewujudkan tujuan dari bisnis tersebut. Suatu model proses bisnis dapat dijalankan oleh satu organisasi atau bisa juga dijalankan berinteraksi dengan proses bisnis organisasi lainnya. Pemodelan proses bisnis dapat bertindak sebagai cetak biru (*blueprint*) rangkaian contoh aktivitas proses bisnis, sedangkan pemodelan aktivitas dapat bertindak sebagai cetak biru beberapa rangkaian aktivitas. Pemodelan proses bisnis terdiri dari beberapa aktivitas yang memiliki batasan implementasi di antara keduanya (Weske, 2007).

2.7.1 Analisis dan Pemodelan Proses Bisnis *As-Is*

Analisis merupakan kegiatan untuk mengumpulkan data sedangkan pemodelan proses bisnis *as-is* merupakan kegiatan untuk memodelkan proses bisnis yang terjadi saat ini, yang memiliki tujuan untuk menggambarkan informasi mengenai aktivitas pada situasi saat ini, mencari tahu kekurangan dari model proses bisnis, dan dapat menemukan perubahan kemungkinan peningkatan dalam model proses bisnis yang ada (Becker, et al., 2013).

Untuk mengidentifikasi proses bisnis saat ini (*as-is*) dapat dilakukan dengan metode wawancara terhadap pegawai yang mengeksekusi proses bisnis tersebut. Proses bisnis saat ini (*as-is*) dapat juga diidentifikasi melalui kegiatan observasi lapangan terhadap pegawai yang sedang mengerjakan tugas-tugasnya, hasil identifikasi proses bisnis *as-is* dapat digunakan menjadi skema dasar pembuatan diagram proses bisnis *as-is* (Przybytek, 2011).

Dalam kasus ini wawancara dilakukan kepada anggota komisi III DPM UB karena komisi tersebut yang menangani terhadap pelayanan aspirasi mahasiswa.

2.7.2 Analisis dan Pemodelan Proses Bisnis *To-Be*

Analisis dan pemodelan proses bisnis usulan (*to-be*) adalah kegiatan untuk mengeksplorasi potensi perbaikan terhadap proses yang telah dimodelkan dalam proses bisnis saat ini (Becker, et al., 2013).





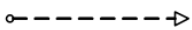
Untuk mendapatkan peningkatan proses bisnis yang mungkin dilakukan pada suatu organisasi/perusahaan yaitu dengan cara melakukan analisis dan

memodelkan proses bisnis usulan (*to-be*) yang melibatkan pemangku kepentingan (Przybytek, 2011). Przybytek menambahkan bahwa peningkatan proses bisnis mungkin dilakukan dengan cara menerapkan otomasi aktivitas dalam proses bisnis. Namun, dalam menerapkan otomasi aktivitas tidak menjamin mendapatkan peningkatan yang sangat signifikan.

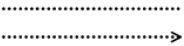



2.7.3 Business Process Modeling Notation (BPMN)

BPMN adalah metodologi yang digunakan untuk memodelkan proses bisnis. Tujuannya untuk memberikan notasi yang dapat mudah dimengerti oleh semua pengguna bisnis, dari bisnis analis yang membuat konsep awal sampai pengembang yang memiliki tanggung jawab dalam penerapan teknologi yang akan menjadikan otomatisasi proses bisnis, dan kepada pelaku yang mengelola serta memantau proses bisnis tersebut (Weske, 2007). Dalam Tabel 2.1 merupakan sebagian penjelasan notasi yang ada dalam BPMN.

Tabel 2.1 Simbol Notasi pada BPMN

No.	Notasi	Nama	Deskripsi
1		<i>Event</i>	<i>Event</i> adalah sesuatu yang terjadi selama proses bisnis berjalan. Event yang mempengaruhi aliran proses dan biasanya memiliki pemicu/hasil. Ada tiga jenis event, yaitu: <i>start</i> , <i>intermediate</i> , dan <i>end</i>
2		<i>Task</i>	<i>Task</i> merepresentasikan aktivitas yang harus dilakukan dalam suatu proses bisnis organisasi/perusahaan
3		<i>Gateway</i>	<i>Gateway</i> merepresentasikan pilihan yang berbeda. Dapat juga merepresentasikan mengenai percabangan dan penggabungan dari <i>flow</i> yang ada. Jenis-jenis <i>gateway</i> : <i>Exclusive</i> , <i>Parallel</i> , <i>Event-based</i> , <i>Complex</i> dan <i>Inclusive</i>
4		<i>Sequence Flow</i>	<i>Sequence flow</i> merepresentasikan mengenai urutan aktivitas yang akan dijalankan dalam satu entitas.
5		<i>Message Flow</i>	<i>Message flow</i> merepresentasikan pertukaran pesan yang

Tabel 2.1 Simbol Notasi pada BPMN (lanjutan)

No.	Notasi	Nama	Deskripsi
			sedang terjadi antar entitas yang berbeda.
6		<i>Association</i>	<i>Association</i> digunakan sebagai asosiasi dari berbagai artifak, data, maupun <i>flow object</i> .
7		<i>Pool</i>	<i>Pool</i> merepresentasikan suatu entitas dalam proses bisnis. <i>Pool</i> biasanya mewakili sebuah organisasi/perusahaan.
8		<i>Lane</i>	<i>Lane</i> merepresentasikan suatu partisi didalam sebuah <i>pool</i> . Biasanya digunakan untuk mewakili unit-unit dalam sebuah organisasi/perusahaan.
9		<i>Data Store</i>	<i>Data Store</i> digunakan untuk memberikan informasi mengenai sebuah lokasi penyimpanan dan pembaruan data.
10		<i>Message</i>	<i>Message</i> merepresentasikan isi komunikasi antar entitas maupun dalam satu entitas.

Sumber: Object Management Group (2011)

2.8 Analisis Persyaratan

Analisis persyaratan merupakan tahapan yang harus dilakukan sebelum melakukan fase perancangan. Analisis persyaratan memiliki tujuan untuk memahami persyaratan sistem yang akan dibangun sehingga dapat memberikan dasar yang akurat dalam perancangan.

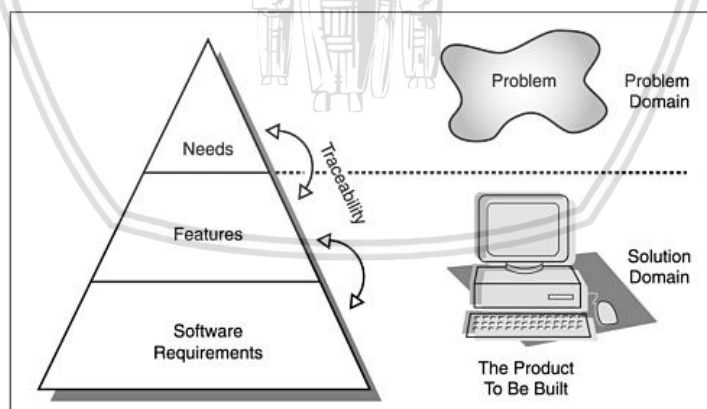
Di dalam analisis persyaratan terdapat beberapa tahapan-tahapan yang dilakukan. Berikut ini adalah tahapan dalam melakukan analisis persyaratan beserta pengertiannya (Sommerville, 2011):

- Elisitasi persyaratan adalah sekumpulan aktivitas yang ditujukan untuk menemukan persyaratan sistem melalui komunikasi antara pengembang dengan pengguna akhir sistem dan pihak lain yang memiliki kepentingan dalam pengembangan sistem. Tujuan dari tahapan elisitasi persyaratan untuk mengetahui domain sistem, fitur yang harus disediakan oleh sistem, kinerja sistem dan sebagainya.
- Spesifikasi persyaratan merupakan tahapan penulisan persyaratan sistem berupa persyaratan pengguna pada sistem dan dituliskan ke dalam

- dokumen persyaratan. Persyaratan pengguna memuat deskripsi persyaratan fungsional dan non fungsional. Persyaratan sistem juga merupakan pengembangan dari persyaratan pengguna yang menjelaskan perilaku eksternal sistem beserta batasan-batasan operasionalnya.
- Validasi dan verifikasi persyaratan adalah tahapan untuk melakukan validasi atau pengecekan kembali terhadap persyaratan-persyaratan sistem yang sudah didefinisikan apakah sudah sesuai berdasarkan keinginan oleh pengguna.
 - Manajemen persyaratan merupakan tahapan dalam mengidentifikasi, memahami, mengontrol, mencari kebutuhan dan perubahan yang terjadi pada persyaratan sistem selama proses pengembangan dijalankan.

Untuk melakukan elisitasi persyaratan diperlukan beberapa tahapan untuk memenuhi kebutuhan nyata dari para pemangku kepentingan. Banyak tim proyek yang terlalu mendalami pemodelan *use case* tanpa mengetahui kebutuhan utama dan kendala yang harus diperbaiki. Memodelkan *use case* tanpa fondasi yang kuat dalam memenuhi kebutuhan utama pemangku kepentingan sering kali menyebabkan masalah yang sangat besar. Untuk menghindari masalah tersebut maka sangat penting untuk mengetahui para pemangku kepentingan, kebutuhan pemangku kepentingan, masalah yang harus diselesaikan dan fitur sistem yang diperlukan pemangku kepentingan (Bittner & Spence, 2002).

Konsep piramida persyaratan menurut Bittner & Spence untuk memperjelas peran, tujuan dan konteks kasus penggunaan. Konsep tersebut menjelaskan bagaimana persyaratan itu bisa ditangkap dan menjelaskan bagaimana persyaratan tersebut mempengaruhi konstruksi detail pemodelan *use case*.



Gambar 2.3 Piramida Persyaratan

Sumber: Bittner & Spence (2002)

2.8.1 Identifikasi Pemangku Kepentingan

Identifikasi pemangku kepentingan merupakan seseorang yang terkena dampak secara materi dari pengembangan suatu sistem (Bittner & Spence, 2002). Tipe pemangku kepentingan menurut Bittner dan Spence terdapat beberapa tipe, yaitu: pengguna, sponsor, pengembang, pihak berwenang dan pelanggan. Mereka

juga menekankan bahwa tipe pemangku kepentingan tidak hanya terbatas pada itu saja, bisa saja dapat dikembangkan tergantung pada jenis perusahaan atau organisasi, macam pengguna sistem, dan lain sebagainya.

2.8.2 Identifikasi Permasalahan Pemangku Kepentingan

Cara terbaik untuk mendapatkan permasalahan pemangku kepentingan yaitu dengan membuat pernyataan masalah (Bittner & Spence, 2002). Adapun pernyataan masalah tersebut dapat diperoleh dengan metode wawancara kepada beberapa pemangku kepentingan, lalu pernyataan masalah tersebut dapat didokumentasikan dalam tabel seperti contoh dalam Tabel 2.2 berikut.

Tabel 2.2 Kerangka Dokumentasi Pernyataan Masalah

<i>The problem of</i>	Berisi deskripsi permasalahan
<i>Affects</i>	Berisi pemangku kepentingan yang terkena dampak permasalahan
<i>The impact of which is</i>	Berisi deskripsi dampak masalah terhadap pemangku kepentingan dan kegiatan bisnis
<i>A successful solution would</i>	Berisi beberapa solusi dan manfaat utama yang ditawarkan dalam penyelesaian permasalahan

Sumber: Bittner & Spence (2002)

2.8.3 Identifikasi Kebutuhan Pemangku Kepentingan

Mengidentifikasi kebutuhan pemangku kepentingan dapat memungkinkan untuk memahami sejauh mana dan bagaimana aspek-aspek yang berbeda dari suatu permasalahan mempengaruhi berbagai jenis pemangku kepentingan yang berbeda (Bittner & Spence, 2002). Mereka juga menyatakan bahwa identifikasi kebutuhan pengguna bisa memberikan pemahaman yang lebih dalam mengenai pernyataan masalah yang disampaikan.

2.8.4 Identifikasi Fitur

Fitur dapat memberikan ringkasan manfaat dari produk yang akan dikembangkan. Dalam memberikan penamaan fitur, harus mempunyai tingkat detail secara umum, sehingga seluruh pemangku kepentingan bisa memahami kemampuan yang ditawarkan sistem, karena fitur juga dapat digunakan untuk meringkas kualitas dan kemampuan dari sebuah aplikasi perangkat lunak yang akan dikembangkan (Bittner & Spence, 2002). Fitur yang sudah ditentukan bisa didokumentasikan ke dalam tabel seperti contoh pada Tabel 2.3 berikut.

Tabel 2.3 Contoh Tabel Fitur

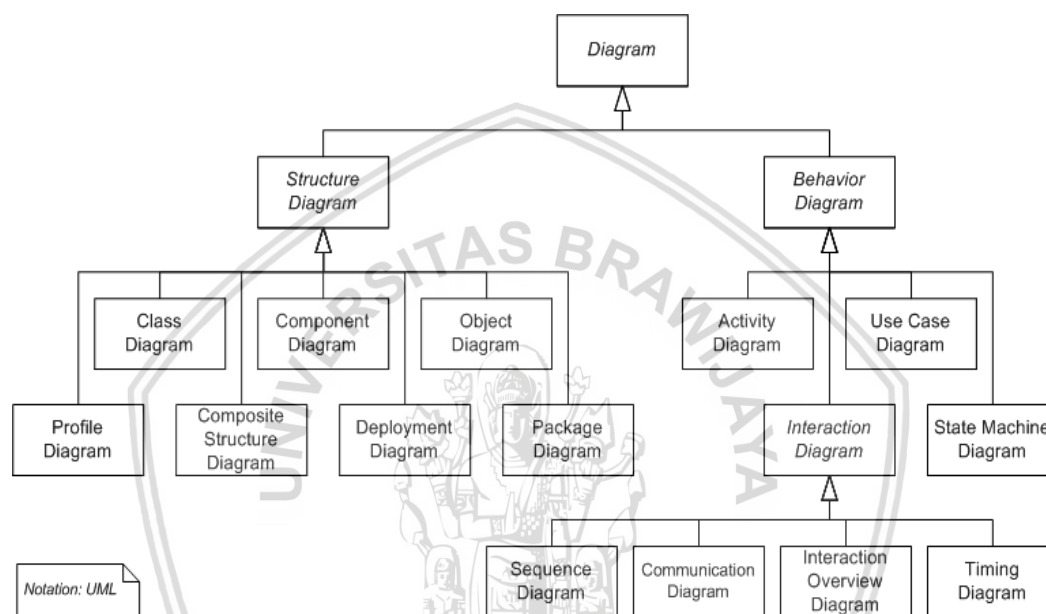
Kode Fitur	Deskripsi
Berisi kode fitur	Berisi penjelasan tentang apa yang bisa dilakukan sistem

Sumber: Bittner & Spence (2002)

2.9 Unified Modelling Language (UML)

UML merupakan suatu bahasa standar dalam penulisan perancangan perangkat lunak. UML juga dapat digunakan untuk memberikan gambaran dalam membangun dan mendokumentasikan artifak yang diperlukan selama proses pengembangan (Pressman, 2010). UML bertujuan untuk menggambarkan suatu mekanisme pemodelan agar dapat dengan mudah dipahami selama proses pembangunan sistem.

Secara hierarki diagram UML dapat dilihat seperti dalam Gambar 2.4 berikut ini.



Gambar 2.4 Hierarki Diagram UML

Sumber: Object Management Group (2011)

2.9.1 Use Case Diagram



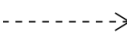


Use case merupakan sebuah diagram yang digunakan untuk mengidentifikasi interaksi individu antara sistem dan penggunaannya atau sistem lain. Setiap *use case* harus didokumentasikan dengan deskripsi tekstual. Diagram ini kemudian dapat dikaitkan dengan model lain di UML yang akan mengembangkan skenario secara lebih rinci (Sommerville, 2011).

Keuntungan menggunakan diagram *use case* adalah mudah dipahami pihak pemangku kepentingan, menunjukan tujuan yang ingin dicapai pengguna dan untuk memastikan fungsional sistem sesuai keinginan pelanggan. Dalam Tabel 2.4 berikut ini merupakan penjelasan dari beberapa simbol diagram *use case*.

Tabel 2.4 Simbol Use Case Diagram

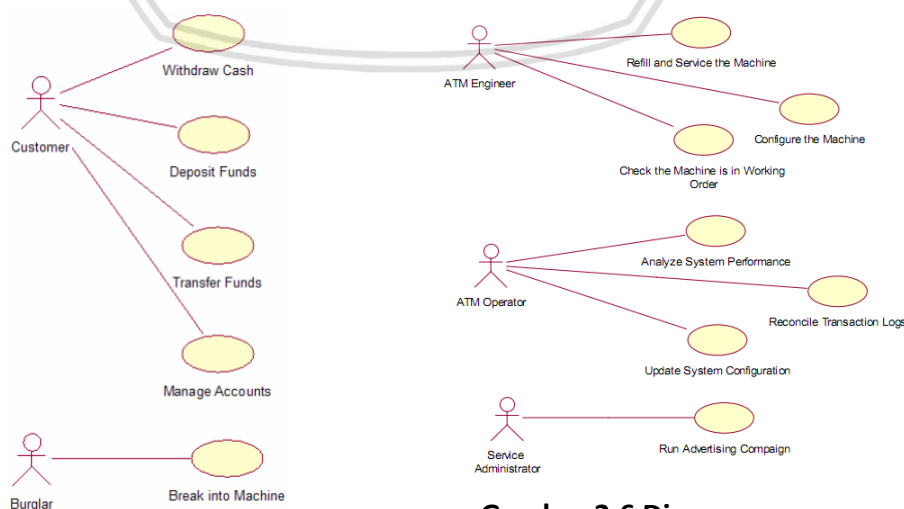
No.	Simbol	Nama	Deskripsi
1		<i>Use Case</i>	Menjelaskan aksi yang dapat dilakukan sistem dan menghasilkan suatu hasil yang terukur bagi aktor

Tabel 2.4 Simbol *Use Case Diagram* (lanjutan)

2		<i>Actor</i>	Menjelaskan tipe peran pengguna saat berinteraksi dengan <i>use case</i>
3		<i>Association</i>	Untuk menghubungkan antar objek satu dengan objek lainnya
4		<i>Dependency</i>	Untuk menghubungkan perubahan yang terjadi pada suatu elemen <i>independent</i> (mandiri) dan akan mempengaruhi ketergantungan elemen yang tidak mandiri
5		<i>Generalization</i>	Untuk menghubungkan objek <i>descendent</i> (anak) berbagi perilaku dengan struktur data objek yang ada di atas objek induk
6		<i>System</i>	Mejelaskan lingkungan sistem yang secara terbatas

Tahapan awal dalam mendesain sistem adalah mengidentifikasi aktor, mengidentifikasi persyaratan fungsionalitas dan membatasi sistem yang akan dikembangkan. Pembatasan sistem ini sangat penting untuk mengidentifikasi aktor, karena dari sinilah kita akan menentukan apakah sesuatu itu adalah suatu aktor dan apakah aktor tersebut dalam bentuk orang atau sistem yang lain. Aktor merupakan segala hal yang berada diluar sistem, yang akan melakukan interaksi dengan sistem tersebut untuk melakukan sesuatu. Jika sudah berhasil mengidentifikasi aktor, maka untuk menemukan *use case* dari sistem akan lebih mudah dilakukan. Karena pada dasarnya, sebuah *use case* harus mendeskripsikan suatu pekerjaan dimana pekerjaan tersebut akan memberikan nilai yang memiliki manfaat bagi aktor (Bittner & Spence, 2002).

Pada Gambar 2.5 dan Gambar 2.6 merupakan contoh proses penulisan *use case* berdasarkan kasus penggunaannya. Berikut ini secara garis besar penggambaran *use case* pada sistem ATM suatu bank (IBM, 2007).

Gambar 2.5 Diagram *use case* utamaGambar 2.6 Diagram *use case* pendukung

Tabel 2.5 berikut ini merupakan contoh dalam mendeskripsikan *use case* menurut Bittner & Spence (2002).

Tabel 2.5 Spesifikasi Use Case

Brief Description	Deskripsi singkat tentang peran dan tujuan dari use case secara umum.
UC Diagram	Merupakan gambaran dari diagram <i>use case</i> .
Preconditions	Deskripsi tekstual untuk mendefinisikan batasan pada sistem ketika <i>use case</i> dapat dimulai.
Basic Flow	Berisikan tentang aliran secara umum dari aktivitas proses bisnis yang ada pada setiap <i>use case</i> .
Alternative Flow	Berisikan informasi tentang kendala atau aliran alternatif yang bisa menyebabkan proses tidak berjalan secara umumnya.
Subflows	Berisikan tentang aliran proses bisnis yang merupakan pecahan dari <i>basic flow</i> sangat kompleks sehingga menjadi suatu proses lain
Postconditions	Deskripsi tekstual yang mendefinisikan kendala pada sistem ketika <i>use case</i> telah dihentikan.
Public Extension Points	Daftar lokasi dalam alur kejadian <i>use case</i> di mana perilaku tambahan dapat dimasukkan.
Special Requirements	Deskripsi tekstual yang mengumpulkan semua persyaratan, seperti persyaratan nonfungsional, pada <i>use case</i> yang tidak dipertimbangkan dalam alur kejadian, tetapi perlu diperhatikan selama desain atau implementasi.

Menurut Tri A. Kurniawan (2018) dalam artikelnya menjelaskan bahwa format dalam mendeskripsikan *use case* yang banyak digunakan dalam praktik hanya bagian-bagian pentingnya saja, yaitu:

1. Aktor Primer (*primary actor*)
2. Prakondisi (*preconditions*)
3. Alur utama (*main or basic flow*)
4. Alur alternatif (*alternative flows*)
5. Kondisi akhir (*postconditions*)

Dalam artikelnya Tri A. Kurniawan juga melakukan identifikasi dan menjelaskan berbagai bentuk kesalahan yang sering terjadi pada saat pembuatan *use case*. Rekomendasi pembetulan yang jelaskan dalam artikel tersebut menjadi acuan dalam pemodelan *use case* agar dapat menghindari kesalahan-kesalahan yang sama dalam proses pengembangan sistem pada penelitian ini, sehingga bisa dihasilkan produk perangkat lunak yang berkualitas dan konsisten.

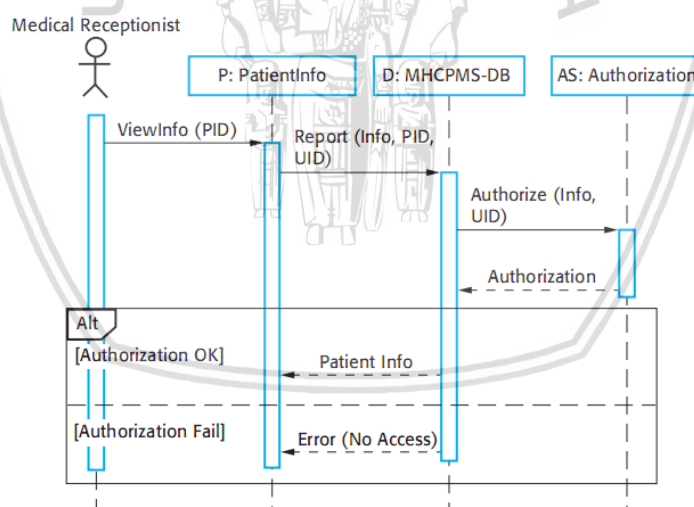
2.9.2 Sequence Diagram

Diagram *sequence* memodelkan urutan dari interaksi aktor dengan objek dan interaksi antar objek yang terjadi dalam suatu sistem. Kelebihan dari diagram *sequence* dapat menunjukkan urutan fungsi sebuah objek dengan objek lain. Diagram *sequence* dapat membantu dalam merepresentasikan secara detail dari suatu *use case* dan dapat digunakan untuk menunjukan interaksi suatu *use case* dalam skenario perangkat lunak (Sommerville, 2011). Tabel 2.6 berikut ini menjelaskan simbol-simbol dari *sequence diagram*.

Tabel 2.6 Simbol Sequence Diagram

No.	Simbol	Nama	Deskripsi
1		<i>LifeLine</i>	Objek entitas antarmuka yang saling berinteraksi/terlibat dalam sistem
2		<i>Object</i>	Elemen yang merepresentasikan jenis apa saja yang berinteraksi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Contoh diagram *sequence* yang didapatkan dari buku Sommerville (2011) terdapat pada Gambar 2.7 berikut.



Gambar 2.7 Contoh Sequence Diagram

Sumber: Sommerville (2011)

2.9.3 Class Diagram

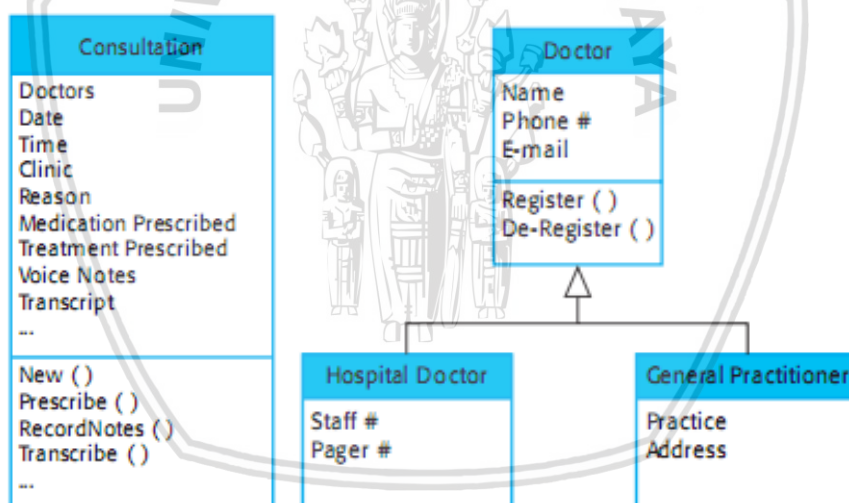
Diagram kelas digunakan ketika mengembangkan sistem dengan pemodelan berorientasi objek yang digunakan untuk menunjukkan kelas dalam suatu sistem dan asosiasi antar masing-masing kelas (Sommerville, 2011).

Berikut ini dalam Tabel 2.7 merupakan beberapa elemen penting dalam sebuah diagram kelas yaitu *class* dan hubungannya.

Tabel 2.7 Simbol *Class Diagram*

No.	Simbol	Nama	Deskripsi
1		<i>Class</i>	Kelas merupakan sebuah template dimana objek dibuat. Kelas mendefinisikan atribut, operasi dan instance.
2		<i>Association</i>	Relasi yang menunjukkan asosiasi/hubungan antar dua kelas.
3		<i>Generalization</i>	Relasi yang menunjukkan pewarisan dari kelas <i>parent</i> (utama) ke kelas <i>child</i> (anak).
4		<i>Dependency</i>	Relasi yang menunjukkan ketergantungan suatu kelas pada kelas lainnya.
5		<i>Aggregation</i>	Relasi yang menunjukkan suatu elemen yang terdiri dari beberapa komponen kecil.

Sebuah kelas diagram dapat menunjukkan entitas berbeda (orang, benda dan data) yang berhubungan satu sama lain dan membentuk struktural atau tampilan statis dari sistem. Pada Gambar 2.8 berikut ini merupakan contoh kelas diagram.

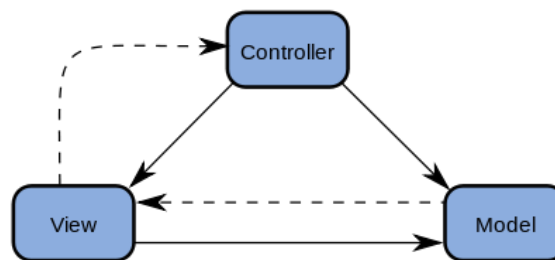


Gambar 2.8 Contoh *Class Diagram*

Sumber: Sommerville (2011)

2.10 Model View Controller (MVC)

MVC merupakan suatu pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasi. Konsep ini dapat meminimalkan *script* dari halaman web seperti *script* presentasi (HTML, JavaScript, CSS, dsb.) dapat dipisahkan dari *script* PHP, dalam istilah umum yaitu untuk menghindari terjadinya *spaghetti code* atau sulit untuk dibaca (Septian, 2011).



Gambar 2.9 Konsep *Model-view-controller*

Dalam Gambar 2.9 merupakan penggambaran konsep MVC yang terdiri dari tiga bagian, yaitu:

a. *Model*

Representasi dari struktur data yang dibangun, pada umumnya kelas model berisi fungsi untuk membantu pengembang dalam mengelola, memasukkan, dan memperbarui informasi dalam basis data.

b. *View*

Representasi informasi yang disajikan pada pengguna berupa *user interface* atau tampilan. *View* merupakan tampilan dari suatu halaman web, namun dalam beberapa *framework*, *view* juga menjadi penggalan-penggalan atau bagian-bagian dari halaman seperti *header* dan *footer*. *View* bisa juga sebagai halaman RSS atau tipe halaman lainnya.

c. *Controller*

Representasi penghubung antara *model*, *view*, dan beberapa sumber lainnya yang diperlukan dalam memproses HTTP *request* untuk menghasilkan suatu halaman web.

2.11 Kuesioner

Kuesioner adalah sejumlah pertanyaan atau pernyataan tertulis tentang opini atau data faktual yang berkaitan dengan responden itu sendiri, yang dianggap fakta atau kebenarannya diketahui dan perlu dijawab oleh responden (Sutoyo, 2009).

Tujuan dari kuesioner adalah untuk mencari jawaban dari sebuah permasalahan yang sedang diteliti. Kuesioner terdiri dari pernyataan serta isian yang harus dijawab oleh responden. Jawaban yang diberikan bisa bersifat tertutup dimana alternatif jawaban telah disediakan oleh peneliti, dan ada juga jawaban terbuka dimana responden bebas menuliskan jawabannya tanpa harus adanya paksaan maupun jawaban yang berasal dari kombinasi keduanya yang merupakan campuran dari jawaban tertutup dan terbuka.

Kuisisioner digunakan dalam penelitian ini untuk mengetahui hasil uji sistem berdasarkan hasil dari beberapa penilaian mahasiswa ataupun operator dan administrator sistem yaitu DPM UB.

2.12 Pengujian

Pengujian perlu dilakukan agar dapat menunjukkan bahwa suatu sistem bisa mengerjakan apa yang diinginkan dan untuk menemukan permasalahan program sebelum digunakan pengguna. Data buatan dapat digunakan dalam menguji perangkat lunak, untuk memeriksa hasil uji coba, anomali, atau informasi tentang atribut non-fungsional sistem (Sommerville, 2011).

Pengujian persyaratan fungsional sistem dapat dilakukan menggunakan metode *whitebox* atau *blackbox* untuk mengetahui apakah sudah berjalan dengan baik atau tidak. Namun, untuk kasus ini hanya menggunakan metode *blackbox* dalam mendapatkan validasi sistem. Sedangkan untuk pengujian non-fungsional sistem, terdapat beberapa yang tidak bisa di uji secara mudah, seperti rentang waktu yang cukup panjang dan kapasitas sistem, untuk menjamin bahwa sistem tidak akan bermasalah maka akan menggunakan pengujian kompatibilitas, penerimaan pengguna dan efisiensi waktu.

Tujuan utama dari pengujian sistem adalah sebagai verifikasi dan validasi sistem untuk membangun kepercayaan bahwa sistem perangkat lunak yang dikembangkan telah tepat sesuai persyaratan dan siap digunakan.

2.12.1 Pengujian *Black Box*

Pengujian *black box* merupakan suatu pendekatan untuk menguji dimana penguji tidak memiliki akses ke kode sumber dari suatu sistem atau komponennya, tes dilakukan berdasarkan dari spesifikasi sistem (Sommerville, 2011).

Validasi sistem dengan menggunakan metode uji *black box*, tidak memerlukan konsentrasi terhadap alur jalannya algoritma program karena lebih menekankan untuk menguji persyaratan fungsional dari sebuah sistem atau perangkat lunak (Pressman, 2010).

2.12.2 Pengujian Validasi

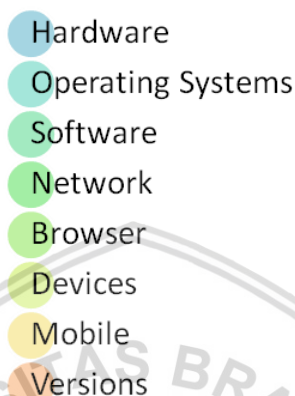
Pengujian validasi adalah pengujian yang berfokus pada tindakan dan hasil keluaran yang terlihat oleh pengguna dari sistem (Pressman, 2010). Pengujian ini dilakukan dengan menggunakan serangkaian kegiatan pengujian, untuk mendemonstrasikan bahwa perangkat lunak telah memenuhi persyaratan awal. Hasil dari pengujian yang dilakukan, menurut Pressman akan mendapatkan dua hasil, yaitu:

1. Karakteristik performa atau fungsi sistem telah memenuhi spesifikasi persyaratan dan diterima oleh pemangku kepentingan atau pelanggan.
2. Deviasi atau kesalahan spesifikasi perangkat lunak dapat terlihat dan kekurangan perangkat lunak juga akan tercatat.

Pengujian validasi sistem dilakukan menggunakan *use case* sebagai panduan pengujian, karena *use case* telah menyediakan skenario yang mungkin untuk mendeteksi deviasi dari sistem yang dikembangkan.

2.12.3 Pengujian Kompatibilitas

Pengujian kompatibilitas adalah jenis pengujian perangkat lunak untuk memeriksa suatu perangkat lunak apakah dapat berjalan pada perangkat keras, sistem operasi, aplikasi, lingkungan jaringan atau perangkat seluler yang berbeda (Guru99, 2018). Jenis lingkungan pengoperasian dalam pengujian kompatibilitas terdapat pada Gambar 2.10 berikut ini.



Gambar 2.10 Jenis-jenis pengujian kompatibilitas

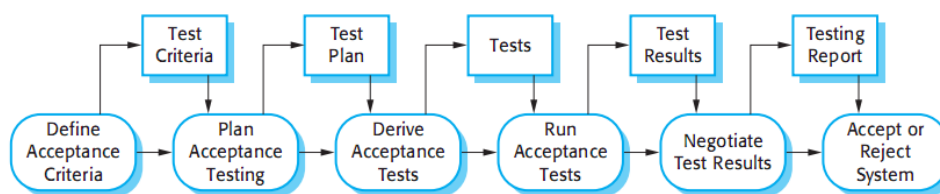
Sumber: Guru99 (2018)

Sistem Informasi Aspirasi *Online* pada DPM UB merupakan perangkat lunak berbasis *web* yang dijalankan dengan menggunakan *web browser*. Sehingga di dalam pengujian kompatibilitas yang diuji adalah lingkungan pengoperasian pada *web browser*. Untuk melakukan pengujian kompatibilitas digunakan perangkat lunak *SortSite*. *SortSite* merupakan aplikasi yang dapat digunakan sebagai alat untuk menjalankan ratusan pemeriksaan kualitas pada setiap halaman situs. *Sortsite* akan mengidentifikasi kesalahan yang terdajadi pada suatu sistem berbasis *web*, seperti *link*, *script* HTML atau format gambar yang rusak/tidak dapat berjalan dan sebagainya. Pada pengujian menggunakan *sortsite*, akan dilakukan uji permasalahan yang sering terjadi pada pengoperasian aplikasi berbasis *website* menggunakan *browser*, yang diantaranya adalah (Guru99, 2018):

1. Kompatibilitas *Internet Explorer*, mulai versi 6.0.
2. Kompatibilitas *desktop browser*, seperti: *Mozilla Firefox*, *Chrome*, *Safari* dan *Microsoft Edge*.
3. Kompatibilitas *mobile browser*, uji *browser* perangkat *mobile*, seperti *Blackberry*, *iPhone/iPad* dan *Android*.

2.12.4 Pengujian Penerimaan Pengguna

User Acceptance Testing (UAT) dilakukan untuk mendapatkan konfirmasi bahwa suatu sistem sudah memenuhi persyaratan yang disepakati bersama (Lau, 2012). Pelanggan menguji sistem untuk memutuskan apakah sistem itu siap atau tidak untuk diterima dari pengembang sistem dan disebarkan ke lingkungan pelanggan (Sommerville, 2011).



Gambar 2.11 Tahapan UAT

Sumber: Sommerville (2011)

Pengujian penerimaan pengguna merupakan bagian tak terpisahkan dari kebiasaan pengujian pengembangan sistem. Pengujian ini terjadi ketika sistem siap dirilis yang melibatkan pelanggan secara formal untuk menguji sistem dan memutuskan apakah sistem diterima atau tidak dari pengembang sistem.

Ada enam tahapan dalam pengujian penerimaan pengguna menurut Sommerville (2011), seperti yang ditunjukkan pada Gambar 2.11, yaitu:

1. *Define acceptance criteria*, merupakan tahapan awal sebelum kontrak ditandatangani. Kriteria penerimaan harus disetujui antara pelanggan dan pengembang.
2. *Plan acceptance testing*, merupakan penentuan sumber daya, waktu dan *budget* dan membuat jadwal tes.
3. *Derive acceptance test*, ketika kriteria penerimaan sudah dibuat, tes harus didesain untuk memeriksa sistem diterima maupun tidak.
4. *Run acceptance test*, tes penerimaan yang telah diterima dan dilakukan pada sistem. Idealnya hal ini harus dilakukan di lapangan dimana sistem tersebut akan digunakan, tapi hal ini cenderung mengganggu dan tidak praktis.
5. *Negotiate test results*, tahap negosiasi antara pelanggan dan pengembang untuk menentukan apakah sistem sudah cukup baik untuk digunakan.
6. *Accept/reject system*, tahapan ini melibatkan pertemuan antara pengembang dan pelanggan untuk memutuskan apakah sistem harus diterima atau tidak. Jika sistem tidak cukup baik untuk digunakan, maka pengembangan lebih lanjut diperlukan untuk memperbaiki masalah yang diidentifikasi. Setelah selesai, fase pengujian penerimaan diulang.

Dalam tahapan *derive acceptance test* terdapat *template* pertanyaan kriteria penerimaan pengguna yang dapat digunakan oleh penguji sebagai rujukan dalam membuat pertanyaan *user acceptance testing*, *template* tersebut didapat dari artikel yang dibuat oleh Jane Lau pada situs web Wiki Singapore Management University (wiki.smu.edu.sg).

PART 1 : REACTIONS TO THE SYSTEM

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1. I found the system's interface (mouse, buttons, icons, and dialogue boxes) an easy way to performing system functions.				2	3
2. I felt secure and comfortable in interacting with the system.				1	4
3. I felt the system responded in a consistent and predictable way.				3	2
4. Please rate the system by checking the word or phrase that most closely reflects your opinion.					
a. information was easy to find				1	4
b. The amount of graphics was appropriate				2	3
c. The terminology was clear and precise			1	2	2
d. I always knew where to access the information I wanted.				5	
e. The amount of explanation was adequate for performing the tasks.				3	2

Gambar 2.12 UAT Feedback Report Template

Sumber: Lau (2012)

Pada Gambar 2.12 merupakan contoh *template* yang telah dijelaskan sebelumnya.

2.12.5 Pengujian Efisiensi Waktu

Efisiensi (*Efficiency*) adalah seberapa cepat pengguna dapat melakukan tugas setelah pengguna mempelajari sistem (Nielsen, 2012). Nielsen menjelaskan beberapa *metric* yang bisa digunakan untuk mengukur efisiensi, yaitu:

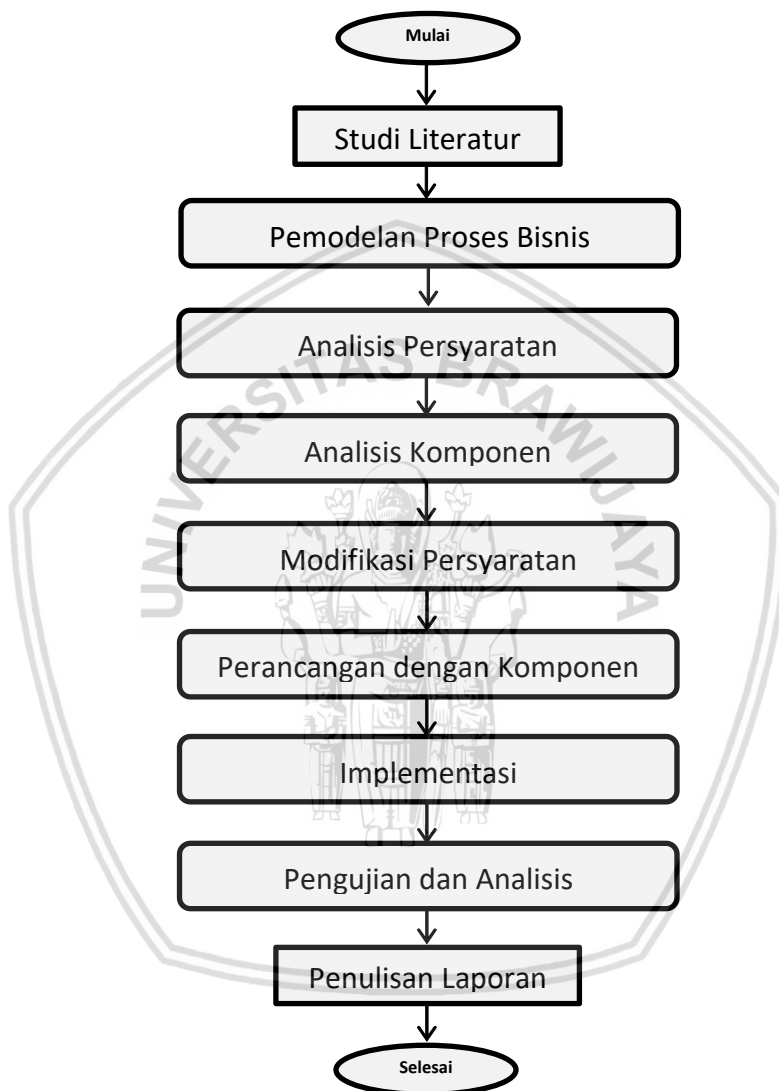
1. Jumlah persentase keberhasilan pengguna dalam menyelesaikan tugas.
2. Waktu yang dibutuhkan dalam melakukan tugas tertentu.
3. Persentase kesalahan dan kepuasan dari sisi subyektif pengguna.

Dari beberapa *metric* yang telah disebutkan sebelumnya, dalam penelitian ini menggunakan *metric* perhitungan waktu yang dibutuhkan dalam melakukan beberapa rangkaian aktivitas yang telah ditentukan. Harapannya, *metric* ini dapat menguji efisiensi waktu dari perbandingan jumlah waktu dalam melakukan aktivitas dengan tanpa menggunakan sistem dan jumlah waktu yang dibutuhkan dalam melakukan tugas dengan menggunakan sistem.

Tujuan dari pengujian ini untuk mengetahui efisiensi waktu yang diperoleh dalam penerapan model proses bisnis usulan (*to-be*) dari model proses bisnis sebelumnya (*as-is*). Perbandingan dari hasil simulasi kedua model proses bisnis tersebut akan membuktikan seberapa besar efisiensi waktu yang didapatkan. Skenario dari pengujian ini adalah dengan cara mengukur waktu aktivitas dari tugas yang diberikan kepada masing-masing responden yang dikelompokkan berdasarkan tipe pengguna.

BAB 3 METODOLOGI PENELITIAN

Pada bab ini, dijelaskan tentang metodologi yang diterapkan selama penelitian skripsi. Adapun tipe penelitian ini adalah penelitian implementatif. Berikut ini merupakan diagram alir dari tahapan-tahapan yang dilakukan dalam penelitian yang gambarkan pada Gambar 3.1.



Gambar 3.1 Diagram Metodologi Penelitian

3.1 Studi Literatur

Pada tahap ini dilakukan untuk mencari literatur dan referensi dari berbagai macam sumber yang didapatkan langsung dari tempat penelitian, jurnal, buku, *e-book*, laporan penelitian dan sumber lainnya serta hasil pencarian internet. Teori pendukung tersebut adalah mengenai proses-proses pengembangan sistem informasi menggunakan pemodelan *reuse-oriented development*. Tujuannya

untuk mendapatkan informasi sebanyak mungkin dan dapat digunakan untuk mendukung penelitian ini.

3.2 Pemodelan Proses Bisnis

Pemodelan proses bisnis digunakan untuk mendefinisikan proses bisnis dan memodelkan aliran aktivitas dan prosedur yang berjalan. Pemodelan proses bisnis dijadikan sebagai acuan dasar untuk mengidentifikasi kebutuhan pengguna yang merupakan bagian utama dalam pengembangan sistem. Untuk itu, dalam pengembangan sistem ini maka diperlukanlah untuk memodelkan proses bisnis pelayanan aspirasi yang terdapat pada DPM UB, sehingga akan dihasilkan beberapa gambaran serangkaian aktivitas yang dilakukan oleh DPM UB ke dalam bentuk diagram proses bisnis yang terbagi menjadi dua kelompok, yaitu proses bisnis yang berjalan saat ini (*as-is*) dan proses bisnis usulan (*to-be*).

Tahapan pemodelan proses bisnis dimulai dengan melakukan observasi dan wawancara dengan anggota DPM UB untuk mengetahui proses bisnis yang berjalan, selanjutnya melakukan analisis proses bisnis usulan sehingga dapat dijadikan acuan untuk menganalisis persyaratan.

3.3 Analisis Persyaratan

Analisis persyaratan merupakan tahapan untuk menspesifikasikan atau menggambarkan dan mendapatkan fungsi yang dibutuhkan secara umum. Dalam tahap analisis persyaratan ini diharapkan mampu memahami bagaimana kebutuhan sistem informasi aspirasi *online* yang akan dibangun. Analisis persyaratan merupakan proses awal dalam menentukan sistem seperti apa yang akan dikembangkan.

Dari hasil pemodelan proses bisnis yang dilakukan sebelumnya akan menghasilkan serangkaian persyaratan sistem berupa kebutuhan fungsional dan non fungsional serta sistem aktor atau pengguna yang terkait dengan sistem. Kebutuhan fungsional sistem secara umum dapat didapatkan sehingga melahirkan persyaratan sistem yang akan dikembangkan. Dari hasil analisis persyaratan maka akan mendapatkan spesifikasi persyaratan (*requirements specification*) yang dibutuhkan dalam pengembangan sistem. Setelah mendapatkan spesifikasi persyaratan selanjutnya akan dimodelkan kedalam bentuk diagram *use case*.

3.4 Analisis Komponen

Analisis komponen (*component analysis*), pada tahapan ini melakukan pengumpulan komponen yang dapat digunakan ulang dan tentunya yang sesuai dengan spesifikasi persyaratan. Analisis komponen dilakukan untuk mendapatkan komponen *reuse* yang sesuai dan dapat diimplementasikan ke dalam sistem yang akan dibuat, biasanya tidak ada komponen yang sepenuhnya tepat dan komponen yang dapat digunakan hanya menyediakan beberapa fungsi saja dari yang diperlukan. Dalam analisis komponen akan membahas daftar komponen-komponen yang dapat digunakan berdasarkan dari hasil analisis persyaratan.

3.5 Modifikasi Persyaratan

Hasil dari analisis komponen pada fase sebelumnya dijadikan sebagai pustaka komponen. Selanjutnya, berdasarkan komponen tersebut modifikasi persyaratan (*requirements modification*) dilakukan agar dapat menyesuaikan dengan komponen. Namun jika tidak memungkinkan untuk dilakukan, analisis komponen dapat dilakukan ulang atau melakukan penambahan, pengurangan, perbaikan dan peningkatan terhadap komponen untuk meminimalisasi waktu pencarian ulang. Fase ini tidak perlu dilakukan apabila komponen sudah sesuai dengan persyaratan.

3.6 Perancangan dengan Komponen

Pada tahap ini, merupakan tahap lanjutan dari tahap analisis komponen dan modifikasi persyaratan. Tahap ini juga termasuk tahapan yang selanjutnya dalam pemodelan *reuse-oriented* karena menghasilkan perancangan sistem dengan penggunaan ulang (*system design with reuse*). Hasil tahapan ini merupakan rancangan berdasarkan komponen *reuse* yang akan diimplementasikan sistem. Dalam perancangan ini dilakukan perancangan kerangka sistem dalam model *sequence diagram*, *class diagram*, *database* dan perancangan antar muka pengguna dalam bentuk *wireframe*.

3.7 Implementasi

Implementasi merupakan suatu tahapan pengembangan, meliputi penerjemahan atau penulisan kode program yang belum ada dan melakukan integrasi komponen-komponen *reuse* dengan menggunakan bahasa pemrograman yang mengacu pada beberapa pemodelan yang telah dibuat pada tahap perancangan, sebagai suatu referensi bagaimana suatu sistem tersebut akan dibentuk dan persyaratan apa saja yang harus terpenuhi dari sistem tersebut.

Pada tahapan ini tidak sepenuhnya mengembangkan dari nol, karena pada tahap sebelumnya telah menyediakan komponen yang dapat digunakan ulang, hanya saja diperlukan integrasi untuk membentuk sistem yang baru.

3.8 Pengujian dan Analisis

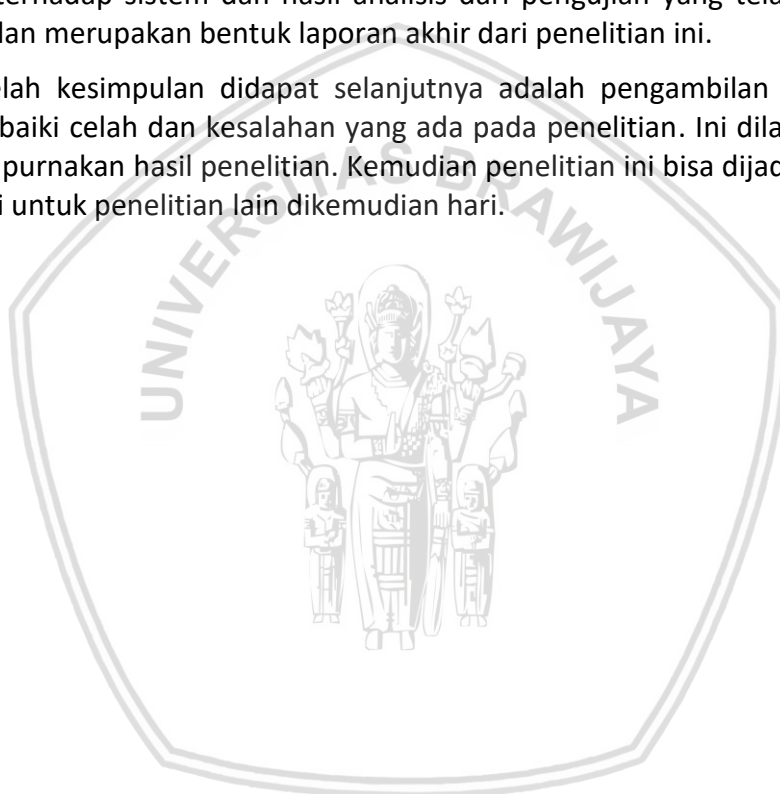
Setelah dilakukan implementasi sistem tahapan selanjutnya adalah pengujian dan analisis, tahap ini diperlukan untuk menjamin bahwa sistem berjalan sesuai dengan harapan, yaitu dengan menguji sistem apakah telah berjalan sesuai dengan spesifikasi persyaratan dan tahapan-tahapan sebelumnya. Pengujian dilakukan menggunakan pengujian validasi dengan metode *blackbox*, pengujian kompatibilitas, pengujian penerimaan pengguna dan pengujian efisiensi waktu. Pengujian validasi dilakukan untuk menguji kebutuhan fungsional sistem apakah sudah dapat berjalan dengan baik dan telah sesuai dengan persyaratan. Pengujian kompatibilitas merupakan pengujian kebutuhan non fungsional yang dilakukan untuk mengetahui *compatibility* dan *accessibility* antar muka sistem pada berbagai *browser*. Pengujian penerimaan pengguna adalah pengujian yang dilakukan pengguna sebagai reponden untuk mencoba sistem dan dapat menilai sistem

melalui kuesioner yang diberikan, pengujian ini dilakukan untuk mengetahui aspek penerimaan pengguna terhadap *usability* dan *security* dalam menggunakan sistem untuk menjawab permasalahan utama pada penelitian ini. Sedangkan pengujian efisiensi waktu merupakan pengujian yang dilakukan untuk membandingkan efisiensi waktu yang didapatkan dari perbandingan waktu model bisnis *as-is* dengan *to-be*.

3.9 Penulisan Laporan

Tahap penulisan laporan tidak mengharuskan menunggu semua tahapan sebelumnya selesai. Namun untuk pengambilan kesimpulan hanya bisa dilakukan setelah tahap sebelumnya sudah selesai dikerjakan. Kesimpulan berisi hasil analisis terhadap sistem dan hasil analisis dari pengujian yang telah dilakukan, kesimpulan merupakan bentuk laporan akhir dari penelitian ini.

Setelah kesimpulan didapat selanjutnya adalah pengambilan saran untuk memperbaiki celah dan kesalahan yang ada pada penelitian. Ini dilakukan untuk menyempurnakan hasil penelitian. Kemudian penelitian ini bisa dijadikan sebagai referensi untuk penelitian lain dikemudian hari.



BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN

Bab ini membahas pemodelan proses bisnis dan analisis persyaratan Sistem Informasi Aspirasi *Online* studi kasus pada DPM UB (SIAP *Online* DPM UB). Pemodelan proses bisnis dan analisis persyaratan diperoleh berdasarkan hasil wawancara dengan anggota DPM UB. Pada bab ini tiga subbab utama meliputi: pemodelan proses bisnis, analisis persyaratan dan pemodelan usecase. Pada tahapan pemodelan proses bisnis terdiri dari pemodelan proses bisnis *as-is* dan pemodelan proses bisnis *to-be*. Untuk tahapan analisis persyaratan terdiri dari identifikasi pemangku kepentingan, permasalahan pemangku kepentingan, kebutuhan pemangku kepentingan, identifikasi pengguna dan fitur. Sedangkan pada tahap pemodelan *use-case* terdiri dari identifikasi aktor, identifikasi *use-case*, pemodelan diagram *use-case* dan spesifikasi *use-case*.

4.1 Pemodelan Proses Bisnis

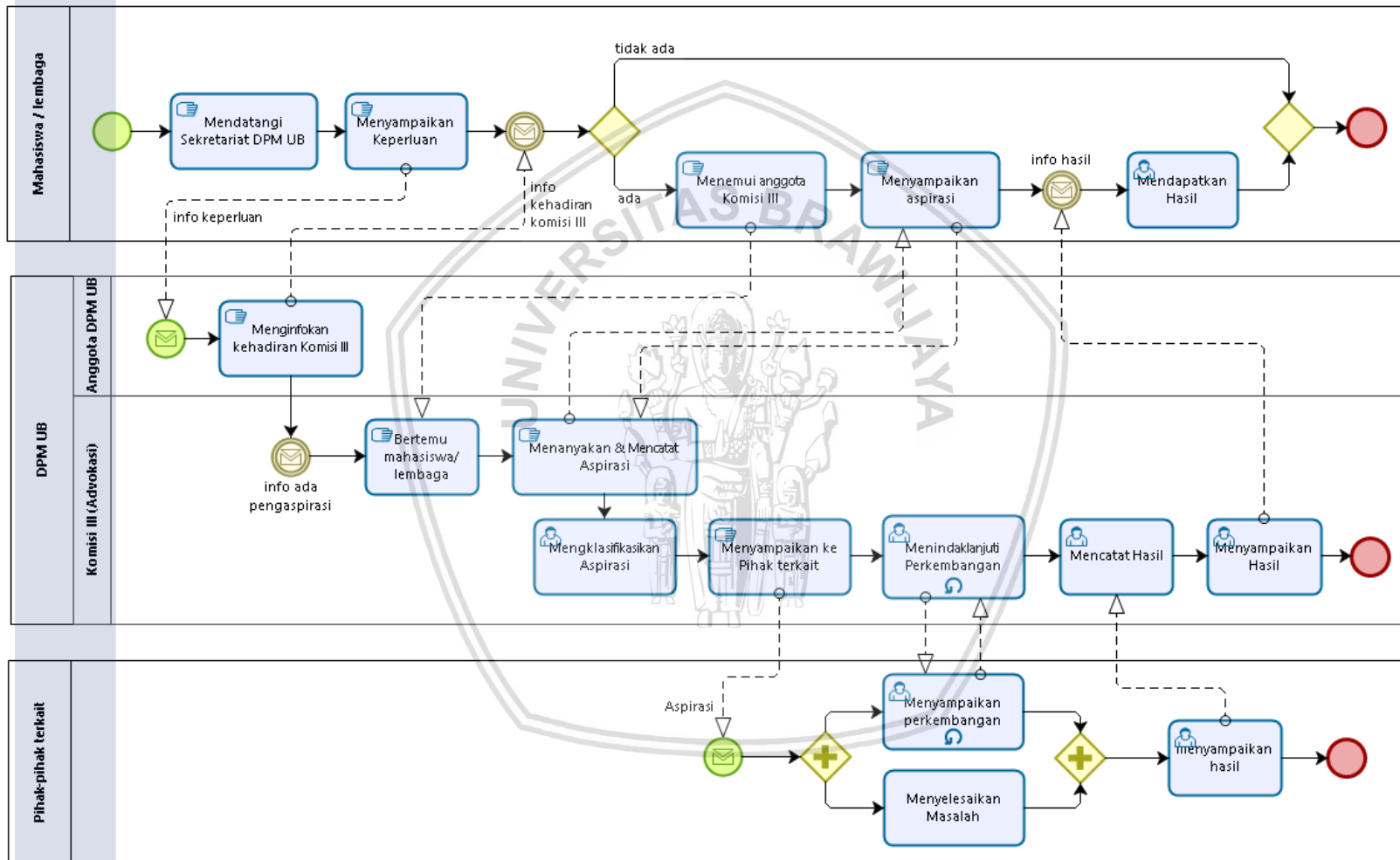
Sebelum masuk ke analisis persyaratan sistem maka diperlukanlah untuk memodelkan proses bisnis pelayanan aspirasi yang terdapat pada DPM UB untuk mendapatkan gambaran dari proses bisnis pada organisasi tersebut, sehingga akan dihasilkan beberapa gambaran serangkaian aktivitas yang dilakukan oleh DPM UB ke dalam bentuk diagram proses bisnis yang terbagi menjadi dua kelompok, yaitu proses bisnis yang saat ini berjalan (*as-is*) dan proses bisnis usulan (*to-be*).

4.1.1 Proses Bisnis *As-Is*

Dalam penelitian ini proses bisnis *as-is* diidentifikasi berdasarkan dari hasil wawancara dan observasi. Proses bisnis ini menggambarkan proses bisnis yang berjalan saat ini pada pelayanan aspirasi mahasiswa di DPM UB. Ada dua cara bagi mahasiswa untuk menyampaikan aspirasinya: Pertama, dimulai dari mahasiswa mendatangi sekretariat DPM UB lalu menemui anggota komisi III. Sedangkan yang kedua, dengan cara mengikuti kegiatan safari fakultas yang diadakan oleh DPM UB lalu bertemu dengan anggota komisi III. Sama halnya dengan DPM UB juga terdapat dua cara untuk menyerap aspirasi mahasiswa: Pertama, mengadakan kegiatan safari fakultas lalu menanyakan permasalahan mahasiswa. Dan yang kedua, menunggu mahasiswa mendatangi sekretariat DPM UB lalu menanyakan permasalahannya.

4.1.1.1 Pelayanan aspirasi melalui sekretariat DPM UB (*as-is*)

Dalam Gambar 4.1 merupakan proses pelayanan aspirasi mahasiswa melalui sekretariat DPM UB. Aktivitasnya dimulai dari mahasiswa yang mendatangi sekretariat, lalu menyampaikan keperluannya kepada anggota DPM, anggota DPM akan mengarahkan ke Komisi III jika anggota Komisi III ada di tempat atau bisa ditemui, namun jika tidak ada maka gagal untuk menyampaikan aspirasinya.



Gambar 4.1 Proses bisnis pelayanan aspirasi melalui sekretariat DPM UB (*as-is*)

Selanjutnya, mereka akan saling bertemu dan anggota komisi III DPM UB menanyakan dan mencatat aspirasi yang disampaikan, anggota DPM UB mencatat dan mendokumentasikan aspirasi mahasiswa kedalam file *excel*. Selanjutnya pengaspirasi harus menunggu sampai hasil didapatkan, aktivitas menunggu hasil tidak mengharuskan berada di sekretariat DPM.

Aspirasi yang tercatat selanjutnya diklasifikasikan terlebih dahulu berdasarkan kategori permasalahan dan tujuannya, lalu DPM menyampaikan ke pihak terkait agar permasalahan tersebut segera diselesaikan.

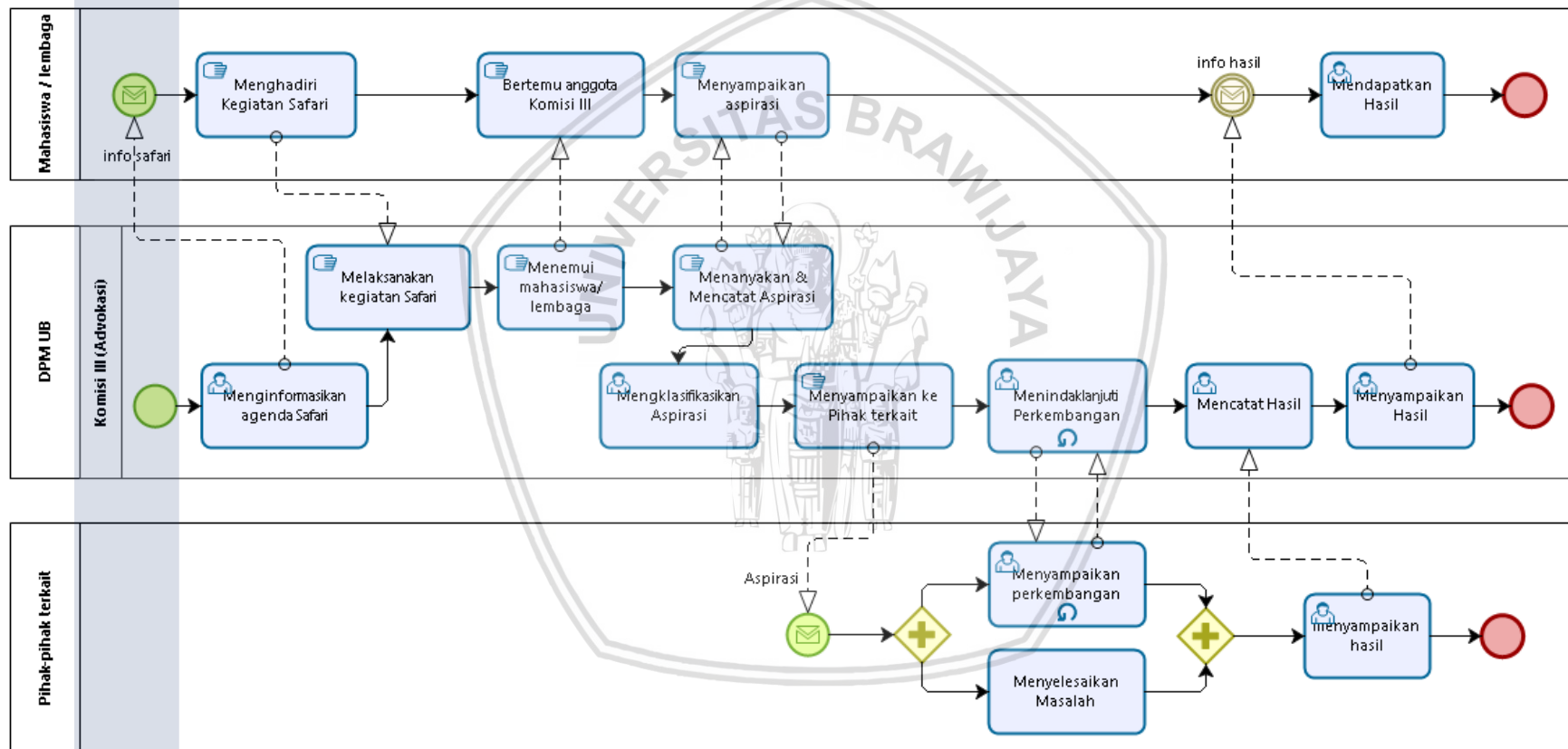
Pihak DPM UB selalu menindaklanjuti perkembangannya dengan menanyakan kepada pihak terkait secara berkala atas aspirasi yang telah disampaikan. Jika aspirasi sudah mendapatkan hasil maka akan dicatat dan disampaikan ke mahasiswa/lembaga yang telah menyampaikan aspirasi tersebut, anggota DPM UB akan menghubungi pengaspirasi tergantung berdasarkan konteks permasalahannya, biasanya disampaikan melalui telepon atau disampaikan dengan tatap muka secara langsung.

4.1.1.2 Pelayanan aspirasi melalui DPM UB Safari Fakultas (as-is)

Safari Fakultas merupakan nama program kerja dari DPM UB itu sendiri, safari fakultas dengan kata lain anggota DPM UB melakukan kegiatan blusukan ke fakultas-fakultas atau lembaga yang ada di Universitas Brawijaya secara terjadwal untuk menyerap aspirasi-aspirasi dari mahasiswa.

Dalam Gambar 4.2 merupakan rangkaian aktivitas pelayanan aspirasi mahasiswa melalui safari fakultas yang dilakukan DPM UB. Prosesnya dimulai dari anggota DPM UB yang menginformasikan agenda kegiatan safari ke fakultas/lembaga tertentu, setelah agenda itu terjadwalkan DPM UB akan melaksanakan kegiatan tersebut dengan cara mengunjungi fakultas/lembaga lalu menemuinya untuk menanyakan aspirasinya dan mahasiswa/lembaga dapat menghadiri kegiatan tersebut lalu bertemu dengan anggota Komisi III untuk menyampaikan aspirasinya. Aspirasinya akan dicatat manual menggunakan file *excel* seperti layanan melalui sekretariat.

Setelah mendapatkan aspirasi, DPM akan mengklasifikasikan aspirasi berdasarkan kategori permasalahannya, lalu DPM UB menyampaikan ke pihak terkait agar permasalahan aspirasi yang disampaikan segera diselesaikan. Proses ini sama dengan layanan melalui sekretariat yaitu pihak komisi III DPM UB akan selalu menindaklanjuti perkembangan dan mencatat hasil sampai mahasiswa mendapatkan hasilnya.



Gambar 4.2 Proses bisnis pelayanan aspirasi melalui DPM UB Safari Fakultas (as-is)

Pada Tabel 4.1 berikut ini merupakan kelebihan dan kekurangan mengenai proses bisnis pelayanan aspirasi secara umum yang berjalan saat ini.

Tabel 4.1 Kelebihan Kekurangan Proses Bisnis Saat Ini (*as-is*)

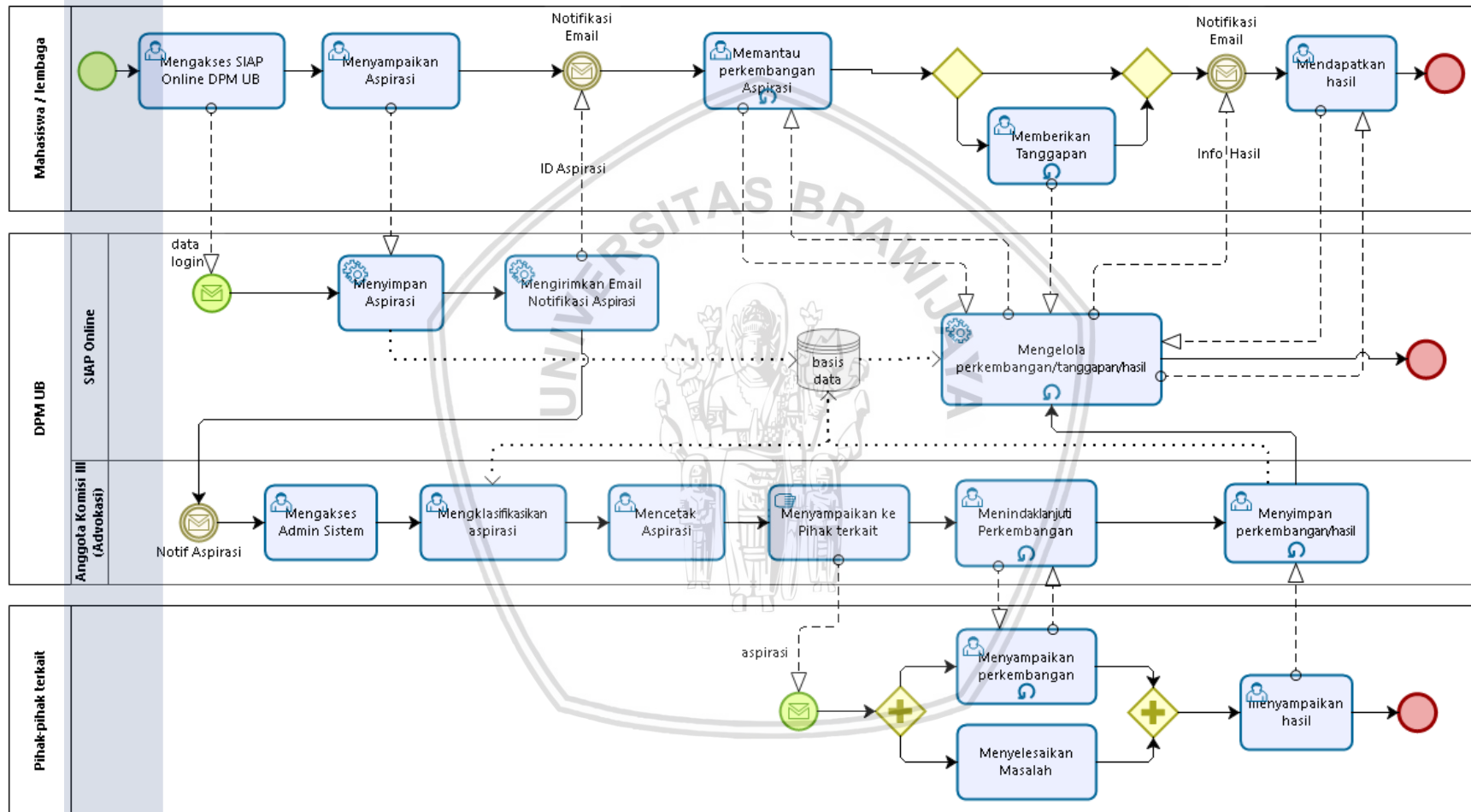
Kekurangan	Kelebihan
<ol style="list-style-type: none"> 1. Mahasiswa/lembaga diharuskan bertemu dan bertatap langsung dengan anggota DPM UB. 2. Pencatatan masih dilakukan secara manual kedalam format <i>excel</i>. 3. Mahasiswa/lembaga tidak bisa memantau perkembangan aspirasi yang disampaikan. 4. Membutuhkan waktu yang lama karena mahasiswa/lembaga harus datang ke sekretariat DPM UB atau menunggu anggota DPM UB melakukan safari fakultas. 5. Belum adanya server basis data yang dapat di akses dari mana saja. 6. Pelayanan menjadi tidak maksimal ketika anggota DPM UB bagian advokasi tidak ada atau tidak dapat ditemui. 	<ol style="list-style-type: none"> 1. DPM UB lebih merasakan permasalahan mahasiswa karena bertemu secara langsung.

4.1.2 Proses Bisnis *To-Be*

4.1.2.1 *Pelayanan aspirasi usulan (to-be)*

Dapat dilihat dalam Gambar 4.3 pada halaman berikutnya, solusi yang dimodelkan dalam model proses bisnis *to-be* adalah dengan memanfaatkan sistem informasi berbasis *online* yang menggunakan basis data sebagai pusat penyimpanannya dan data tersebut dapat dikelola sesuai kebutuhan aktivitas layanan. Solusi ini sebenarnya tidak sepenuhnya menghapus model proses bisnis yang ada saat ini, hanya saja menyediakan layanan model proses bisnis baru yang mengintegrasikan proses penyampaian dan penyerapan aspirasi mahasiswa atau lembaga terhadap DPM UB.

Dalam model proses bisnis *to-be* ini secara tidak langsung mahasiswa/lembaga terhubung dengan anggota DPM UB, dengan dilayani oleh sistem yang akan dikembangkan. Dari proses penyampaian aspirasi sampai mendapatkan hasil dilakukan dan disampaikan otomatis dengan sistem.



Gambar 4.3 Proses bisnis pelayanan aspirasi usulan (*to-be*)

Dengan adanya otomatisasi aktivitas yang dilakukan oleh sistem maka dapat mengurangi aktivitas yang seharusnya dilakukan oleh anggota DPM UB, sehingga beban kinerja anggota DPM UB dapat berkurang. Bahkan, dengan adanya sistem ini tidak mengharuskan selalu adanya anggota DPM UB dalam proses menerima aspirasi yang disampaikan oleh mahasiswa/lembaga karena sistem dapat menyimpannya terlebih dulu dan dapat disampaikan ke pihak terkait ketika anggota DPM UB sudah ada.

Pada Tabel 4.2 berikut ini akan menjabarkan kelebihan dan kekurangan mengenai proses bisnis usulan pelayanan aspirasi pada DPM UB.

Tabel 4.2 Kelebihan Kekurangan Proses Bisnis Usulan (*to-be*)

Kekurangan	Kelebihan
<ol style="list-style-type: none"> 1. Membutuhkan koneksi internet untuk mengakses sistem. 2. Layanan tidak dapat beroperasi jika terjadi <i>server down</i>. 	<ol style="list-style-type: none"> 1. Mahasiswa/lembaga dapat menyampaikan aspirasinya secara <i>online</i> tanpa harus bertemu dengan anggota DPM UB. 2. Pelayanan aspirasi tak terbatas oleh waktu karena sistem <i>online</i> selama 24 jam. 3. Mempercepat proses penyampaian karena tidak mengharuskan datang ke sekretariat DPM atau mengikuti kegiatan safari fakultas. 4. Otomatisasi penyimpanan data aspirasi mahasiswa ke dalam basis data sehingga tidak memerlukan tenaga anggota DPM UB lagi. 5. Klasifikasi aspirasi dapat dilakukan menggunakan sistem secara otomatis. 6. Mahasiswa/lembaga dapat memantau dan menerima notifikasi perkembangan, tanggapan dan hasil dari aspirasinya melalui sistem. 7. Mahasiswa dapat memberikan tanggapan/<i>feedback</i> atas aspirasinya.

Berikut ini, pada Tabel 4.3 merupakan beberapa perubahan yang terjadi pada aktivitas pelayanan aspirasi di DPM UB.

Tabel 4.3 Perubahan aktivitas proses bisnis pelayanan apirasi secara umum

Unit Bisnis	Proses Bisnis As-Is	Proses Bisnis To-Be	Keterangan
Mahasiswa/lembaga	Mendatangi sekretariat DPM UB	-	Dieliminasi
	Menghadiri Kegiatan Safari Fakultas	-	Dieliminasi
	Menyampaikan Keperluan	-	Dieliminasi
	Menemui anggota komisi III DPM UB	-	Dieliminasi
	-	Mengakses SIAP <i>Online</i> DPM UB**	Ditambah
	Menyampaikan aspirasi*	Menyampaikan aspirasi**	Diubah
	-	Memantau perkembangan **	Ditambah
	-	Memberikan tanggapan**	Ditambah
Anggota DPM UB	Menginfokan kehadiran komisi III	-	Dieliminasi
Komisi III (Advokasi)	Menginformasikan agenda Safari Fakultas	-	Dieliminasi
	Melaksanakan kegiatan Safari Fakultas	-	Dieliminasi
	Menemui mahasiswa/lembaga	-	Dieliminasi
	Menanyakan & Mencatat aspirasi*	Menyimpan aspirasi***	Diubah
	-	Mengakses Admin Sistem**	Ditambah
	Mengklasifikasikan aspirasi*	Mengklasifikasikan aspirasi**	Diubah
	-	Mencetak Aspirasi**	Ditambah
	Mencatat hasil*	Menyimpan perkembangan/hasil**	Diubah
	Menyampaikan Hasil	Mengelola perkembangan/ tanggapan/hasil***	Diubah
Pihak-pihak terkait	-	-	-

Keterangan: *) dilakukan secara manual, **) dilakukan menggunakan sistem
 ***) dilakukan otomatis oleh sistem.

4.2 Analisis Persyaratan

Analisis persyaratan sistem terdiri dari elisitasi kebutuhan, identifikasi aktor, spesifikasi kebutuhan, verifikasi dan validasi. Tahapan-tahapan di dalam proses analisis persyaratan bertujuan untuk mendapatkan semua persyaratan yang diperlukan oleh sistem yang akan dibangun. Proses dalam mendefinisikan persyaratan sistem dilakukan pada tahap elisitasi kebutuhan yang menggunakan teknik wawancara dan observasi kepada anggota komisi III bagian advokasi DPM UB. Berikut hasil analisis persyaratan sistem informasi aspirasi *online* pada DPM UB.

4.2.1 Elisitasi Persyaratan

Pada tahap elisitasi persyaratan, pengembang bekerja sama dengan pengguna akhir dari sistem yang akan dikembangkan untuk mengetahui domain aplikasi, fitur apa saja yang harus disediakan sistem, kinerja yang diperlukan sistem, kendala *hardware* dan sebagainya (Sommerville, 2011). Pada penelitian ini tahap elisitasi kebutuhan dilakukan dengan metode wawancara dan observasi. Metode wawancara dan observasi dipilih karena dapat berinteraksi langsung dengan *stakeholder* sistem yang merupakan bagian besar dari proses analisis persyaratan untuk merumuskan persyaratan-persyaratan sistem. Dalam melakukan elisitasi persyaratan dapat dibagi menjadi empat tahapan, yaitu: Identifikasi tipe pemangku kepentingan, Identifikasi kebutuhan pemangku kepentingan, identifikasi pengguna dan identifikasi fitur.

4.2.1.1 Identifikasi pemangku kepentingan

Identifikasi pemangku kepentingan dilakukan untuk mengidentifikasi berbagai tipe pemangku kepentingan yang dilibatkan dalam proyek. Identifikasi pemangku kepentingan harus mencakup informasi berupa nama tipe pemangku kepentingan dan deskripsi singkat mengenai tipe pemangku kepentingan yang berhubungan dengan sistem atau proyek (Bittner & Spence, 2002).

Hasil identifikasi tipe pemangku kepentingan adalah daftar pemangku kepentingan yang telah dikelompokkan berdasarkan karakteristik dan hubungan pemangku kepentingan pada sistem yang akan dikembangkan, hasil ini digunakan untuk mengidentifikasi pengguna terhadap sistem. Tabel 4.4 berikut ini merupakan para pemangku kepentingan yang terlibat dalam proses pengembangan SIAP *Online* DPM UB.

Tabel 4.4 Hasil identifikasi pemangku kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Pengguna	Individu yang berinteraksi dengan sistem informasi secara langsung yang berperan sebagai aktor pada <i>use case</i> .	1. Mahasiswa
		2. Lembaga

Tabel 4.4 Hasil identifikasi pemangku kepentingan (lanjutan)

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
		3. Staff ahli/anggota komisi III
		4. Ketua komisi III
Pengembang	Organisasi atau individu yang melakukan pengembangan SIAP <i>Online</i> DPM UB.	Peneliti
Pihak-pihak yang berwenang	Organisasi atau individu yang memiliki kemampuan dalam mendukung pengembangan dengan cara memberikan informasi mengenai regulasi yang berjalan dan solusi yang akan dikembangkan dapat berjalan sesuai dengan aturan yang berlaku.	DPM UB
Pihak-pihak yang terkait	Organisasi atau individu yang terkena dampak sistem secara tidak langsung terkait dalam penanganan permasalahan aspirasi mahasiswa/lembaga.	1. Civitas akademika Universitas Brawijaya
		2. Anggota LKM UB

4.2.1.2 Identifikasi Permasalahan pemangku kepentingan

Identifikasi permasalahan pemangku kepentingan dilakukan untuk memahami masalah yang terjadi saat ini dan harus diselesaikan oleh pemangku kepentingan. Beberapa permasalahan yang sebelumnya telah diuraikan di latar belakang penelitian ini akan didetailkan pada tahapan ini.

Pada tabel 4.5 berikut ini merupakan permasalahan dari sisi penyampai aspirasi yaitu mahasiswa dan lembaga.

Tabel 4.5 Pernyataan masalah pertama

<i>The problem of</i>	<ol style="list-style-type: none"> 1. Banyaknya mahasiswa yang kurang peduli/apatis terhadap lembaga kemahasiswaan yang ada di UB. 2. Mahasiswa terkendala harus datang ke sekretariat DPM untuk menyampaikan aspirasinya
------------------------------	---

Tabel 4.5 Pernyataan masalah pertama (lanjutan)

	3. Mahasiswa harus bertatap muka langsung dengan anggota DPM UB bagian advokasi 4. Aspirasi yang masuk sangat kurang
Affects	Pengguna, pihak-pihak yang berwenang, dan pihak-pihak terkait.
The impact of which is	Fungsi dan tujuan LKM UB menjadi tidak maksimal.
A successful solution would	Sistem yang dapat menyediakan layanan penyampaian aspirasi dan aduan secara <i>online</i> yang dapat diakses kapan pun dan dimana pun.

Selanjutnya masuk ke permasalahan yang kedua, yang dilihat dari sisi DPM UB sebagai penyedia layanan atau yang menerima aduan/aspirasi mahasiswa dan lembaga. DPM UB sebagai lembaga legislatif yang mempunyai fungsi pengawasan dan mengadvokasikan permasalahan yang terjadi, maka membutuhkan data aspirasi dari mahasiswa/lembaga untuk dijadikan sebagai bukti atau landasan dalam mengambil keputusan. Namun DPM UB mengalami permasalahan dalam pelayanannya. Tabel 4.6 berikut ini akan mendetailkan permasalahan yang dialami oleh DPM UB.

Tabel 4.6 Pernyataan masalah kedua

The problem of	1. Pelayanan yang diberikan DPM UB kurang maksimal 2. DPM UB lambat dalam menyerap aspirasi mahasiswa 3. Anggota DPM UB tidak mampu menemui semua elemen-elemen mahasiswa secara langsung 4. Pencatatan data aspirasi masih dilakukan manual menggunakan <i>excel</i> yang terkadang menyebabkan hilangnya data aspirasi
Affects	Pengguna, pihak-pihak yang berwenang, dan pihak-pihak terkait.
The impact of which is	Data aspirasi yang dibutuhkan untuk dijadikan acuan dalam melakukan advokasi pada pihak-pihak terkait menjadi lemah dan kurang.
A successful solution would	Sistem yang dapat menyediakan layanan penyerapan dan pengolahan data aspirasi secara terpusat.

4.2.1.3 Identifikasi kebutuhan pemangku kepentingan

Kebutuhan pemangku kepentingan berisi sekumpulan pernyataan yang berhubungan dengan solusi atas kekurangan yang terjadi pada pelayanan aspirasi pada DPM UB, sebagaimana telah dipaparkan pada sub-bab pemodelan proses bisnis. Hasil identifikasi kebutuhan pemangku kepentingan digunakan sebagai dasar informasi untuk menganalisis fitur yang dibutuhkan oleh pengguna. Tahapan ini merupakan tahapan untuk mengidentifikasi solusi yang ditawarkan dalam memenuhi kebutuhan. Masing-masing kebutuhan pemangku kepentingan diberi kode sebagai identitas, kode pada setiap kebutuhan pemangku kepentingan ditandai dengan kode SIAP-PK-00. SIAP merupakan singkatan dari Sistem Informasi Aspirasi, PK singkatan dari Pemangku Kepentingan dan angka 00 sebagai penomoran kebutuhan pemangku kepentingan.

Dari hasil wawancara dan observasi yang telah dilakukan, maka didapatkan temuan-temuan kebutuhan pemangku kepentingan yang menjadi pengguna sistem. Kebutuhan tersebut dijabarkan pada Tabel 4.7 berikut ini.

Tabel 4.7 Identifikasi kebutuhan pemangku kepentingan

Kode Kebutuhan	Kebutuhan	Pemangku kepentingan	Situasi saat ini	Solusi yang ditawarkan
SIAP-PK-01	Dapat membuktikan identitas mahasiswa/ lembaga yang akan menyampaikan aspirasi	Mahasiswa, Lembaga	Pembuktian identitas menggunakan kartu tanda mahasiswa (KTM)	Sistem menggunakan otentikasi mahasiswa dengan akun yang digunakan untuk <i>login</i> SIAM
SIAP-PK-02	Menyediakan layanan penyampaian aspirasi	Mahasiswa, Lembaga	Penyampaian aspirasi dilakukan secara manual dan harus tatap muka langsung dengan anggota komisi III	Sistem menyediakan penyampaian aspirasi secara <i>online</i> yang dapat diisi oleh mahasiswa/ lembaga
SIAP-PK-03	Mencatat aspirasi yang disampaikan	Staff ahli/anggota komisi III,	Dilakukan dengan mencatat secara	Sistem menyimpan ke dalam basis

Tabel 4.7 Identifikasi kebutuhan pemangku kepentingan

Kode Kebutuhan	Kebutuhan	Pemangku kepentingan	Situasi saat ini	Solusi yang ditawarkan
		ketua komisi III	manual ke dalam <i>file excel</i>	data secara otomatis
SIAP-PK-04	Mengklasifikasi data aspirasi berdasarkan permasalahan dan tujuannya	Staff ahli/anggota komisi III, ketua komisi III	Dilakukan dengan mengurutkan data yang telah tersimpan lalu memisahnya melalui fitur disediakan oleh <i>excel</i>	Sistem dapat mengelola data sesuai keinginan pemangku kepentingan termasuk mengurutkan dan memisahkan secara otomatis
SIAP-PK-05	Melakukan <i>follow-up</i> (menanyakan) status perkembangan aspirasinya	Mahasiswa, Lembaga	Tidak ada	Sistem menyediakan fitur untuk memantau perkembangan atau tanggapan atas aspirasi yang disampaikan
SIAP-PK-06	Mencatat perkembangan /hasil yang didapat dari pihak-pihak terkait	Staff ahli/anggota komisi III, ketua komisi III	Dilakukan dengan mencatat secara manual ke dalam <i>file excel</i>	Sistem dapat menyimpan status perkembangan /hasil kedalam basis data
SIAP-PK-07	Memberikan tanggapan (<i>feedback</i>) terhadap aspirasi yang telah disampaikan	Mahasiswa, Lembaga, Staff ahli/anggota komisi III, ketua komisi III	Tidak ada	Sistem menyediakan tempat interaksi dalam menanggapi aspirasi

Tabel 4.7 Identifikasi kebutuhan pemangku kepentingan (lanjutan)

Kode Kebutuhan	Kebutuhan	Pemangku kepentingan	Situasi saat ini	Solusi yang ditawarkan
SIAP-PK-08	Memberikan pemberitahuan terkait perkembangan/ tanggapan/ hasil dari suatu aspirasi	Mahasiswa, Lembaga, Staff ahli/anggota komisi III, ketua komisi III	Dilakukan dengan bertemu langsung atau melalui telepon	Sistem dapat menyampaikan perkembangan aspirasi melalui notifikasi yang dikirim oleh sistem
SIAP-PK-09	Memberikan layanan yang mudah digunakan bagi mahasiswa/ lembaga dan bagi DPM itu sendiri	Mahasiswa, Lembaga, Staff ahli/anggota komisi III, ketua komisi III	Pelayanan melalui sekretariat DPM UB dan melalui Safari Fakultas	Sistem dapat menjadi layanan baru dalam pengelolaan aspirasi

4.2.1.4 Identifikasi pengguna

Identifikasi pengguna adalah tahapan yang dilakukan untuk mengetahui individu yang akan menggunakan sistem secara langsung. Pengguna merupakan perwujudan dari sebagian pemangku kepentingan. Pengguna sistem merupakan bagian dari kelompok kategori pengguna dan pengembang dalam tipe pemangku kepentingan. Informasi mengenai identifikasi pengguna digunakan untuk melakukan identifikasi fitur dan identifikasi aktor yang akan dimodelkan ke dalam diagram *use case*. Pada Tabel 4.8 berikut ini merupakan hasil dari identifikasi pengguna.

Tabel 4.8 Hasil Identifikasi Pengguna

Tipe Pemangku Kepentingan	Tipe Pengguna	Keterangan
Pengguna	Mahasiswa	Seseorang yang dapat menyampaikan dan menerima hasil aspirasi/aduan.
	Lembaga	Perwakilan Unit Aktivitas Mahasiswa, EM dan DPM UB yang dapat menyampaikan dan menerima hasil aspirasi/aduan.
	Operator	Staff ahli dan anggota komisi III DPM UB yang dapat menjawab atau memberikan tanggapan

Tabel 4.8 Hasil Identifikasi Pengguna (lanjutan)

Tipe Pemangku Kepentingan	Tipe Pengguna	Keterangan
		terhadap aspirasi/aduan dan menyampaikan ke pihak-pihak terkait.
	Administrator	Ketua komisi III DPM UB yang memiliki kedudukan tertinggi dan bertanggungjawab penuh terhadap segala hal pelayanan aspirasi mahasiswa termasuk dalam pengelolaan aspirasi, akun lembaga dan operator.
Pengembang	Peneliti	Pengguna yang memiliki akses dalam pengembangan sistem untuk mengelola data dan perbaikan terhadap sistem.

4.2.1.5 Identifikasi Fitur

Fitur merupakan representasi kemampuan beberapa solusi dari sistem yang ditawarkan untuk memberikan manfaat bagi pengguna dan membantu pemenuhan kebutuhan pemangku kepentingan. Fitur menyediakan ringkasan manfaat dari produk yang akan dibangun, fitur bisa fungsional atau non-fungsional (Bittner & Spence, 2002).

Masing-masing fitur diberikan kode sebagai identitas, kode pada setiap fitur ditandai dengan kode F00. F merupakan representasi dari kata fitur, sedangkan angka 00 sebagai penomoran dari fitur.

Hasil dari identifikasi fitur merupakan spesifikasi kebutuhan dari pengguna yang akan digunakan untuk menspesifikasikan persyaratan fungsional dan non-fungsional sistem informasi yang akan dikembangkan. Hasil identifikasi fitur terdapat pada Tabel 4.9.

Tabel 4.9 Hasil Identifikasi Fitur

Kode Fitur	Deskripsi
F01	Sistem dapat mengenali identitas pengguna sistem (mahasiswa/lembaga) dan membatasi akses pengguna terhadap informasi dan fungsi-fungsi yang disediakan oleh sistem berdasarkan hak akses pengguna.
F02	Sistem dapat digunakan untuk mengirimkan aspirasi/aduan dengan cara mengisi data pada sebuah <i>form</i> yang memuat identitas lengkap, subyek, kategori, isi aspirasi/aduan dan lampiran berupa foto (jika ada) dan menyimpannya ke basis data terpusat.

Tabel 4.9 Hasil Identifikasi Fitur (lanjutan)

Kode Fitur	Deskripsi
F03	Sistem dapat mengirimkan notifikasi email yang memberitahukan ID aspirasi dan dapat mengirimkan notifikasi balasan jika aspirasi mendapatkan tanggapan.
F04	Sistem dapat digunakan untuk mengelola data aspirasi termasuk mengklasifikasikan data, merekap data dan mengekspor data kedalam format <i>excel</i> .
F05	Sistem dapat digunakan untuk melihat statistik dari data aspirasi yang diterima.
F06	Sistem dapat digunakan untuk menyimpan status perkembangan, tanggapan dan/atau hasil dari aspirasi yang disampaikan.
F07	Sistem dapat digunakan untuk memberikan tanggapan/balasan pada setiap aspirasi yang disampaikan.
F08	Sistem dapat digunakan untuk memantau perkembangan, tanggapan dan/atau hasil dari aspirasi yang disampaikan.
F09	Sistem dapat digunakan untuk mengelola akun lembaga dan operator agar bisa <i>login</i> sistem.
F10	Sistem dapat digunakan untuk memisahkan data-data aspirasi yang tidak penting (<i>spam dll</i>).
F11	Sistem dapat digunakan dengan mudah, memiliki tampilan <i>user friendly</i> dan <i>responsive</i> .

Masing-masing fitur pada penelitian ini akan dikelompokkan berdasarkan keterkaitan terhadap kebutuhan pemangku kepentingan yang menjadi pengguna sistem, karena kebutuhan tersebut harus dipenuhi oleh sistem untuk menyelesaikan permasalahannya. Pengelompokan ini menggambarkan bahwa sebuah fitur merupakan solusi untuk memenuhi kebutuhan tertentu. Pengelompokan fitur dengan kebutuhan pemangku kepentingan terdapat pada Tabel 4.10.

Tabel 4.10 Hubungan kebutuhan pemangku kepentingan dengan fitur

Kode kebutuhan pemangku kepentingan	Kode fitur
SIAP-PK-01	F01
SIAP-PK-02	F02
SIAP-PK-03	F02, F03
SIAP-PK-04	F04, F05, F10

Tabel 4.10 Hubungan kebutuhan pemangku kepentingan dengan fitur (lanjutan)

Kode kebutuhan pemangku kepentingan	Kode fitur
SIAP-PK-05	F08
SIAP-PK-06	F06
SIAP-PK-07	F07, F09
SIAP-PK-08	F08
SIAP-PK-09	F11

4.2.2 Spesifikasi persyaratan

Spesifikasi persyaratan adalah proses menuliskan jenis pengguna dan persyaratan sistem ke dalam dokumen persyaratan. Idealnya, pengguna dan persyaratan sistem harus jelas, tidak ambigu, mudah dimengerti, lengkap, dan konsisten. Namun dalam praktiknya, sulit untuk dicapai karena para pemangku kepentingan menafsirkan persyaratan dengan cara yang berbeda dan sering ada konflik yang inheren dan inkonsistensi dalam persyaratan (Sommerville, 2011). Sommerville juga menjelaskan bahwa persyaratan sistem merupakan pengembangan dari fitur pengguna yang menjelaskan perilaku eksternal sistem beserta batasan-batasan operasionalnya. Fitur pengguna terhadap sistem memuat deskripsi persyaratan fungsional dan non fungsional yang diuraikan pada tahapan berikut ini.

4.2.2.1 Persyaratan Fungsional

Persyaratan fungsional merupakan persyaratan fungsi yang terdapat pada sistem, layanan yang harus disediakan sistem, bagaimana reaksi sistem terhadap suatu masukan data dan apa yang harus dilakukan sistem pada situasi tertentu (Sommerville, 2011). Tahapan ini akan menjelaskan persyaratan fungsional berdasarkan pengguna yang berinteraksi terhadap sistem beserta dengan kode, nama dan deskripsi fungsi dari masing-masing persyaratan.

Pada tahapan ini dilakukan identifikasi fungsi dari sistem, fungsi tersebut didapatkan dari hasil mendetailkan fitur sistem yang sudah diidentifikasi pada tahapan sebelumnya. Dari fungsi inilah didapatkan spesifikasi persyaratan fungsional sistem.

Kode pada setiap spesifikasi persyaratan fungsional ditandai dengan kode PF-01-2. PF merupakan singkatan dari Persyaratan Fungsional dan angka 01 sebagai penomoran persyaratan utama dan angka 2 adalah persyaratan spesifik yang menjelaskan persyaratan utama.

Berikut ini merupakan persyaratan fungsional dan spesifikasi persyaratan fungsional sistem informasi aspirasi *online* pada DPM UB yang dijabarkan pada Tabel 4.11.

Tabel 4.11 Persyaratan Fungsional

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
F01	PF-01	Login	<p>Sistem harus menyediakan fungsi <i>login</i> bagi <i>guest</i> untuk diidentifikasi hak aksesnya.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> (PF-01-1) Sistem harus membedakan <i>login</i> mahasiswa/lembaga dengan <i>login</i> operator (PF-01-2) Sistem menyediakan <i>form login</i> dengan pilihan sebagai mahasiswa/lembaga dan masukan berupa NIM/<i>username</i> dan kata sandi (PF-01-3) Sistem tidak dapat memproses <i>login</i> apabila inputan NIM/<i>username</i> atau kata sandi masih kosong (PF-01-4) Sistem menampilkan pesan kesalahan apabila NIM/<i>username</i> atau kata sandi tidak sesuai (PF-01-5) Sistem melakukan otentikasi mahasiswa menggunakan API <i>bais.ub.ac.id</i> (PF-01-6) Sistem melakukan otentikasi lembaga dan operator melalui <i>username</i> yang sudah didaftarkan oleh administrator
	PF-02	Logout	<p>Sistem harus menyediakan fungsi <i>logout</i> bagi pengguna.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> (PF-02-1) Sistem dapat mengakhiri sesi dari pengguna yang sedang <i>login</i>
F02	PF-03	Melengkapi identitas	<p>Sistem harus menyediakan fungsi melengkapi identitas bagi pengguna baru.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> (PF-03-1) Sistem dapat melengkapi identitas mahasiswa yang didapatkan

Tabel 4.11 Persyaratan Fungsional

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
			<p>dari API <i>bais.ub.ac.id</i> yaitu NIM, nama, fakultas, jurusan dan program studi. Sehingga perlu menambah kontak yang dapat dihubungi yaitu nomor HP dan email.</p> <p>2. (PF-03-2) Sistem harus dapat memverifikasi email yang didaftarkan</p> <p>3. (PF-03-3) Sistem tidak dapat menyimpan data dan tidak dapat lanjut ke fitur lainnya jika data identitas belum dilengkapi</p>
	PF-04	Memperbarui identitas	<p>Sistem harus menyediakan fungsi memperbarui identitas bagi pengguna yang sudah melengkapi identitas.</p> <p>Spesifikasi persyaratan:</p> <p>1. (PF-04-1) Sistem dapat memperbarui identitas kontak (<i>email</i> dan nomor <i>handphone</i>) bagi mahasiswa/lembaga</p> <p>2. (PF-04-2) Sistem harus dapat memverifikasi email yang didaftarkan</p> <p>3. (PF-04-3) Sistem tidak dapat menyimpan data jika kontak tidak lengkap</p>
	PF-05	Mengirimkan aspirasi	<p>Sistem harus menyediakan fungsi mengirimkan aspirasi bagi mahasiswa dan lembaga.</p> <p>Spesifikasi persyaratan:</p> <p>1. (PF-05-1) Sistem dapat mengirimkan aspirasi yang menerima masukan berupa subyek, kategori, isi aspirasi/aduan dan lampiran berupa foto.</p> <p>2. (PF-05-2) Sistem menyediakan kategori yang dapat dipilih berdasarkan jenis permasalahannya:</p>

Tabel 4.11 Persyaratan Fungsional

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
			<p>permasalahan pribadi, permasalahan lembaga dan permasalahan global.</p> <p>3. (PF-05-3) Sistem tidak dapat menyimpan data aspirasi jika data belum lengkap</p>
F03	PF-06	Mendapatkan email notifikasi	<p>Sistem harus menyediakan fungsi notifikasi yang dikirimkan melalui email.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> 1. (PF-06-1) Sistem dapat mengirimkan ID aspirasi melalui <i>email</i> yang berisi ID aspirasi kepada pengguna 2. (PF-06-1) Sistem dapat mengirimkan notifikasi aspirasi melalui <i>email</i> jika ada tanggapan atas aspirasinya
F04	PF-07	Mengelola daftar aspirasi	<p>Sistem harus menyediakan fungsi mengelola daftar aspirasi bagi operator.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> 1. (PF-07-1) Sistem dapat menampilkan daftar aspirasi dalam bentuk tabel, yang menampilkan tanggal, subjek dan status aspirasi 2. (PF-07-2) Sistem dapat melakukan klasifikasi data berdasarkan <i>user</i>, status, tujuan atau kategori 3. (PF-07-3) Sistem dapat melakukan pencarian aspirasi 4. (PF-07-4) Sistem dapat melakukan cetak (<i>print</i>) detail aspirasi 5. (PF-07-5) Sistem dapat melakukan ekspor data aspirasi kedalam format <i>excel</i>
F05	PF-08	Melihat statistik aspirasi	<p>Sistem harus menyediakan fungsi untuk melihat statistik aspirasi bagi operator.</p> <p>Spesifikasi persyaratan:</p>

Tabel 4.11 Persyaratan Fungsional

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
			<ol style="list-style-type: none"> 1. (PF-08-1) Sistem dapat menampilkan statistik aspirasi berdasarkan fakultas, <i>user</i> (mahasiswa/lembaga), kategori dan bulan 2. (PF-08-2) Sistem dapat menampilkan statistik dalam bentuk <i>chart</i>
F06	PF-09	Mencatat perkembangan aspirasi	<p>Sistem harus menyediakan fungsi untuk mencatat perkembangan aspirasi bagi operator.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> 1. (PF-09-1) Sistem dapat menyimpan perkembangan data aspirasi yang didapatkan dari pihak-pihak terkait 2. (PF-09-2) Perkembangan dalam bentuk status: belum proses, diterima, ditindaklanjuti dan selesai
F07	PF-10	Memberikan tanggapan aspirasi	<p>Sistem harus menyediakan fungsi untuk memberikan tanggapan pada suatu aspirasi bagi pengguna.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> 1. (PF-10-1) Sistem dapat mengirimkan dan menyimpan tanggapan pada setiap aspirasi yang disampaikan 2. (PF-10-2) Tanggapan hanya bisa dilakukan apabila sesi dalam keadaan <i>login</i> (pengguna sistem)
F08	PF-11	Memantau perkembangan aspirasi	<p>Sistem harus menyediakan fungsi untuk memantau perkembangan suatu aspirasi.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> 1. (PF-11-1) Sistem dapat melakukan lacak (<i>tracking</i>) aspirasi melalui ID aspirasi lalu menampilkan detail aspirasi 2. (PF-11-2) Sistem dapat menampilkan status perkembangan aspirasi

Tabel 4.11 Persyaratan Fungsional

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
			3. (PF-11-3) Sistem dapat melakukan cetak (<i>print</i>) detail aspirasi 4. (PF-11-3) Sistem dapat menampilkan kolom tanggapan
F09	PF-12	Membuat akun lembaga	Sistem harus menyediakan fungsi bagi administrator untuk membuat akun baru bagi lembaga yang belum mempunyai akun. Spesifikasi persyaratan: <ol style="list-style-type: none"> (PF-12-1) Sistem dapat menambah akun lembaga dengan menerima masukan berupa: <i>username</i>, <i>password</i>, nama lembaga, nomor <i>handphone</i> dan <i>email</i> (PF-12-2) Sistem menampilkan pesan kesalahan apabila terdapat masukan yang masih kosong
	PF-13	Menghapus akun lembaga	Sistem harus menyediakan fungsi bagi administrator untuk menghapus akun lembaga. Spesifikasi persyaratan: <ol style="list-style-type: none"> (PF-13-1) Sistem dapat menampilkan tabel daftar akun lembaga (PF-13-2) Sistem memberikan peringatan terlebih dulu sebelum menghapus akun lembaga (PF-13-3) Ketika akun dihapus tidak menghapus data aspirasi dari akun tersebut
	PF-14	Membuat akun operator	Sistem harus menyediakan fungsi bagi administrator untuk membuat akun baru bagi operator. Spesifikasi persyaratan: <ol style="list-style-type: none"> (PF-14-1) Sistem dapat menambah akun operator dengan menerima masukan berupa: <i>username</i>,

Tabel 4.11 Persyaratan Fungsional

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
F10			<p><i>password</i>, nama lembaga, nomor <i>handphone</i> dan <i>email</i></p> <p>2. (PF-14-2) Sistem menampilkan pesan kesalahan apabila terdapat masukan yang masih kosong</p>
	PF-15	Menghapus akun operator	<p>Sistem harus menyediakan fungsi bagi administrator untuk menghapus akun operator.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> 1. (PF-15-1) Sistem dapat menampilkan tabel daftar akun operator 2. (PF-15-2) Sistem memberikan peringatan terlebih dulu sebelum menghapus akun operator 3. (PF-15-3) Ketika akun dihapus tidak menghapus pada data tanggapan aspirasi yang pernah dilakukan akun tersebut
	PF-16	Menghapus aspirasi	<p>Sistem harus menyediakan fungsi bagi administrator untuk menghapus aspirasi sehingga tidak tampil di data operator.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> 1. (PF-16-1) Sistem tidak sepenuhnya menghapus melainkan hanya memindahkan ke tong sampah 2. (PF-16-2) Sistem memberikan peringatan terlebih dulu sebelum menghapus aspirasi
	PF-17	Mengelola tong sampah	<p>Sistem harus menyediakan fungsi bagi administrator untuk mengelola data aspirasi yang ada di tong sampah.</p> <p>Spesifikasi persyaratan:</p> <ol style="list-style-type: none"> 1. (PF-17-1) Sistem dapat menampilkan daftar aspirasi yang sudah dihapus ke dalam bentuk tabel

Tabel 4.11 Persyaratan Fungsional (lanjutan)

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
			2. (PF-17-2) Sistem dapat mengembalikan data yang berada di tong sampah ke tempah semula

4.2.2.2 Persyaratan Non Fungsional

Persyaratan non fungsional merupakan fungsi yang berhubungan dengan perilaku sistem dalam lingkup operasional yang dapat berupa keandalan sistem, keamanan, kinerja dan sifat sistem lainnya yang dimunculkan (Sommerville, 2011). Kode yang digunakan pada setiap persyaratan non fungsional sama dengan kode kebutuhan fungsional. Hanya mengganti kode PF yang singkatan dari persyaratan fungsional menjadi PNF yang merupakan kepanjangan dari persyaratan non fungsional. Pada Tabel 4.12 berikut ini mendeskripsikan kebutuhan non fungsional sistem informasi aspirasi *online* pada DPM UB.

Tabel 4.12 Persyaratan Non Fungsional

Kode fitur	Kode	Parameter	Deskripsi
F01	PNF-01	<i>Security</i>	Sistem membatasi hak akses pengguna dengan memberikan hak akses sesuai dengan jenis akun pengguna.
F11	PNF-02	<i>Usability</i>	Sistem dapat digunakan dengan mudah dan menampilkan fitur-fitur sesuai dengan kebutuhan pengguna dengan tampilan <i>user friendly</i> .
	PNF-03	<i>Compatibility</i>	Sistem dapat berjalan di berbagai <i>browser</i> yang berbeda (termasuk <i>browser smartphone</i>).
	PNF-04	<i>Accessibility</i>	Tampilan sistem dapat menyesuaikan berdasarkan ukuran <i>device</i> dengan menerapkan fitur <i>responsive</i> pada desain sistem.

4.2.3 Verifikasi dan Validasi

Verifikasi dan validasi persyaratan adalah tahapan untuk pengecekan kembali terhadap persyaratan-persyaratan sistem yang sudah didefinisikan apakah sudah sesuai berdasarkan keinginan pengguna (Sommerville, 2011). Proses verifikasi dilakukan untuk memeriksa apakah persyaratan sudah memenuhi kebutuhan yang ditetapkan. Sedangkan proses validasi dilakukan untuk memastikan apakah persyaratan sistem sudah memenuhi harapan pengguna. Proses verifikasi dan

validasi pada penelitian ini dilakukan oleh bagian advokasi komisi III DPM UB yang terdapat dalam Lampiran B Verifikasi dan Validasi Persyaratan yang hasilnya dapat dilihat pada Tabel 4.13 dan Tabel 4.14 berikut ini.

Tabel 4.13 Verifikasi dan Validasi Persyaratan Fungsional

Kode Fitur	Kode Fungsi	Nama Fungsi	Verifikasi	Validasi
F01	PF-01	Login	√	√
	PF-02	Logout	√	√
F02	PF-03	Melengkapi identitas	√	√
	PF-04	Memperbarui identitas	√	√
	PF-05	Mengirimkan aspirasi	√	√
F03	PF-06	Mendapatkan email notifikasi	√	√
F04	PF-07	Mengelola daftar aspirasi	√	√
F05	PF-08	Melihat statistik aspirasi	√	√
F06	PF-09	Mencatat perkembangan aspirasi	√	√
F07	PF-10	Memberikan tanggapan aspirasi	√	√
F08	PF-11	Memantau perkembangan aspirasi	√	√
F09	PF-12	Membuat akun lembaga	√	√
	PF-13	Menghapus akun lembaga	√	√
	PF-14	Membuat akun operator	√	√
	PF-15	Menghapus akun operator	√	√
F10	PF-16	Menghapus aspirasi	√	√
	PF-17	Mengelola tong sampah	√	√

Tabel 4.14 Verifikasi dan Validasi Persyaratan Non Fungsional

Kode fitur	Kode	Parameter	Verifikasi	Validasi
F01	PNF-01	Security	√	√
F11	PNF-02	Usability	√	√
	PNF-03	Compatibility	√	√
	PNF-04	Accessibility	√	√

4.3 Pemodelan Use Case

Pemodelan *use case* ini dilakukan setelah mengetahui hasil dari analisis persyaratan. Pada tahapan ini dilakukan identifikasi aktor, identifikasi *use case* dan spesifikasi skenario dari masing-masing *use case*. Pemodelan *use case* pada penelitian ini berpedoman pada buku Bittner & Spence (2002) dan mempelajari artikel dari Tri A. Kurniawan (2018) yang berjudul “*Pemodelan Use Case (UML): Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik*” untuk mengurangi kesalahan yang akan terjadi dalam praktek pemodelannya.

4.3.1 Identifikasi Aktor

Aktor merupakan orang atau sistem yang berasal dari eksternal sistem yang dapat menerima dan menyediakan informasi terhadap sistem. Identifikasi aktor didapatkan dari hasil identifikasi pengguna yang dilakukan pada tahapan elisitasi persyaratan. Namun, dari hasil tersebut belum melibatkan aktor lain yang juga terlibat dengan sistem. Karena secara definisi aktor itu tidak hanya sebatas pengguna saja, melainkan sistem lain yang berhubungan dengan sistem juga merupakan aktor. Tabel 4.15 berikut ini merupakan hasil identifikasi aktor yang berinteraksi dengan SIAP Online DPM UB.

Tabel 4.15 Identifikasi Aktor

No	Aktor	Deskripsi
1	BAIS	Sistem eksternal yang menyediakan <i>web service</i> untuk otentikasi akun mahasiswa Universitas Brawijaya melalui API.
2	Visitor	Merupakan pengunjung yang sedang mengakses sistem. Visitor dapat melakukan <i>login</i> dan memantau aspirasi menggunakan ID aspirasi yang telah diberikan.
3	Mahasiswa/Lembaga	Mahasiswa/lembaga merupakan aktor yang dapat mengirimkan aspirasi, mengelola identitasnya dan memberikan balasan terhadap tanggapan dari operator.
4	Operator	Operator bertanggung jawab untuk memberikan tanggapan terhadap aspirasi yang disampaikan, mengelola data aspirasi dan dapat melihat statistik aspirasi.
5	Administrator	Administrator memiliki tanggung jawab yang sama seperti operator dan bertanggung jawab untuk mengelola akun, menghapus aspirasi yang tidak penting dan memindahkannya ke dalam tong sampah.

4.3.2 Identifikasi *Use Case*

Use case adalah interaksi antara aktor dengan sistem untuk melakukan sesuatu yang berguna (Bittner & Spence, 2002). *Use case* merupakan gambaran berdasarkan persyaratan fungsional sistem yang telah didefinisikan sebelumnya. Namun, fungsi pada persyaratan fungsional belum tentu *use case*. Bittner & Spence menambahkan agar tidak bingung membedakan antara *use case* dengan fungsi, coba dipikirkan kembali dengan menggunakan pertanyaan “Apakah semua *use case* ini bermanfaat secara independen?” jika tidak, berarti itu hanya sekedar fungsi atau menu dari sebuah sistem.

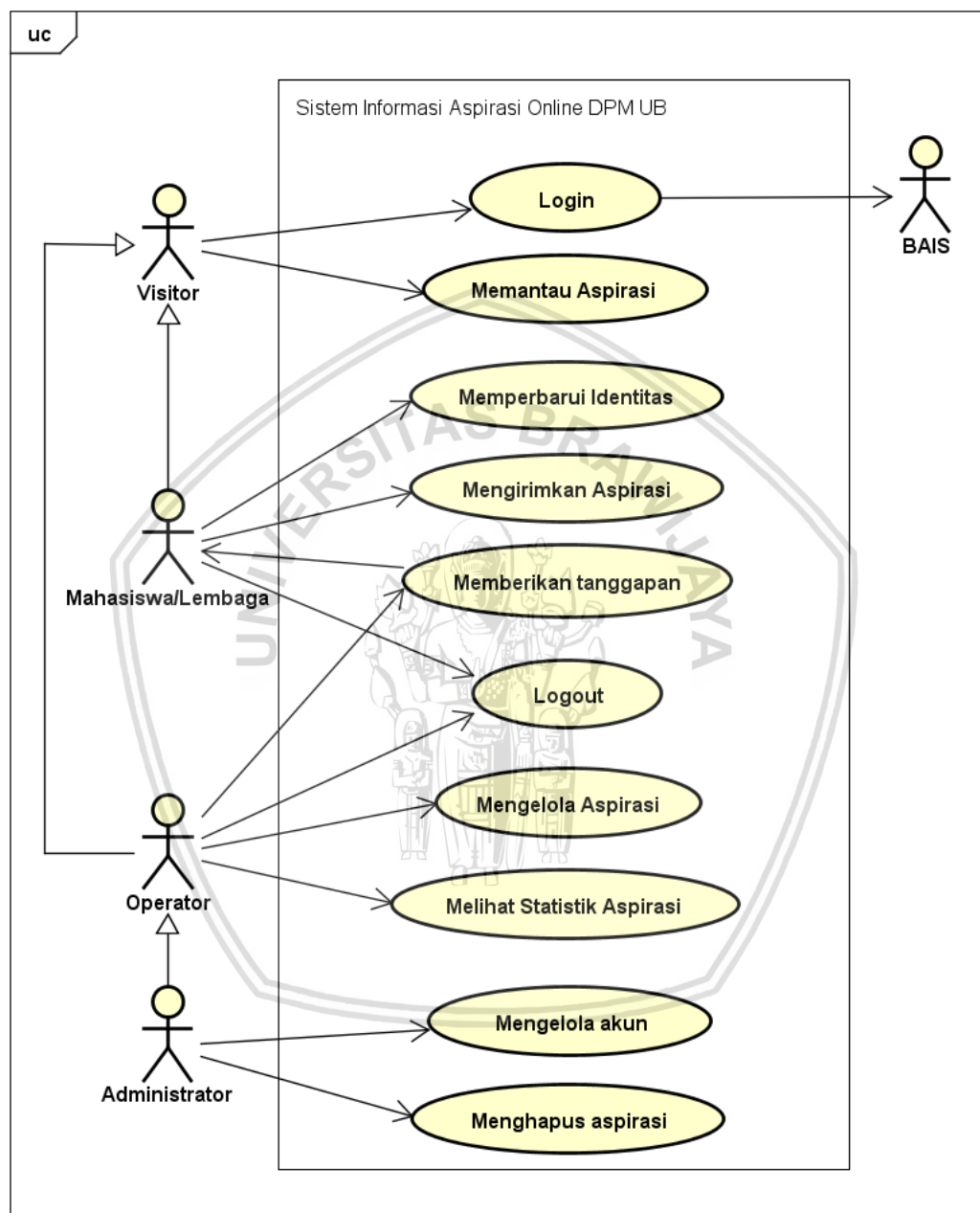
Pada Tabel 4.16 berikut ini, merupakan hasil dari identifikasi *use case* dan hubungannya dengan persyaratan fungsional sistem yang telah teridentifikasi sebelumnya. Ini dilakukan untuk menunjukkan bahwa *use case* sudah sesuai dengan persyaratan fungsional sistem yang dibutuhkan oleh pemangku kepentingan.

Tabel 4.16 Hubungan *Use Case* dengan Persyaratan Fungsional

Use Case		Persyaratan fungsional	
Kode	Nama	Kode	Nama
UC-01	Login	PF-01	Login
UC-02	Logout	PF-02	Logout
UC-03	Memperbarui Identitas	PF-03	Melengkapi identitas
		PF-04	Memperbarui identitas
UC-04	Mengirimkan aspirasi	PF-05	Mengirimkan aspirasi
		PF-06	Mendapatkan email notifikasi
UC-05	Mengelola aspirasi	PF-07	Mengelola daftar aspirasi
UC-06	Melihat statistik aspirasi	PF-08	Melihat statistik aspirasi
UC-07	Memberikan tanggapan	PF-09	Mencatat perkembangan aspirasi
		PF-10	Memberikan tanggapan aspirasi
UC-08	Memantau aspirasi	PF-11	Memantau perkembangan aspirasi
UC-09	Mengelola akun	PF-12	Membuat akun lembaga
		PF-13	Menghapus akun lembaga
		PF-14	Membuat akun operator
		PF-15	Menghapus akun operator
UC-10	Menghapus aspirasi	PF-16	Menghapus aspirasi
		PF-17	Mengelola tong sampah

4.3.3 Pemodelan Diagram *Use Case*

Pada Gambar 4.4 berikut merupakan hasil pemodelan diagram *use case* dari Sistem Informasi Aspirasi *Online* pada DPM UB. Tanda panah pada *use-case* menunjukkan siapa yang melakukan inisiasi terlebih dahulu.



Gambar 4.4 Diagram *Use Case* SIAP *Online* DPM UB

4.3.4 Spesifikasi *Use Case*

Spesifikasi *use case* merupakan penjelasan singkat mengenai aktivitas dari setiap *use case* yang telah teridentifikasi. Spesifikasi juga memuat informasi mengenai aktor yang menjalankan *use case*, kondisi sebelum dan sesudah *use case* dijalankan, serta alur/skenario yang dilakukan ketika *use case* jalankan.

4.3.4.1 Spesifikasi Use Case Login

Tabel 4.17 menjelaskan tujuan dan bagaimana *visitor* dapat masuk ke dalam sistem. Di dalam spesifikasi *use case login* terdapat alur kondisi sistem untuk mengenali aktor yang sedang menggunakan sistem, sehingga sistem dapat memberikan batasan dan tujuan sesuai haknya.

Tabel 4.17 Spesifikasi Use Case Login

Code	UC-01
Primary Actor	Visitor
Brief Description	<i>Use case Login</i> menjelaskan bagaimana aktor visitor dapat menggunakan sistem sesuai dengan hasil verifikasi identitasnya, sehingga aktor dapat mengakses informasi pada sistem sesuai dengan hak aksesnya.
Pre-conditions	Aktor harus akses fungsi <i>login</i> pada sistem.
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika sistem menampilkan <i>form</i> isian data yang terdiri dari tipe pengguna, NIM/<i>username</i> dan <i>password</i>. 2. Aktor mengisi <i>form</i> secara lengkap kemudian memerintahkan sistem untuk memproses. <p>{mengidentifikasi tipe}</p> <ol style="list-style-type: none"> 3. Sistem mengidentifikasi tipe sebagai mahasiswa. <p>{otentikasi mahasiswa}</p> <ol style="list-style-type: none"> 4. Sistem melakukan otentikasi menggunakan <i>web service</i> yang disediakan oleh BAIS untuk mendapatkan identitas mahasiswa. 5. Sistem memberikan hak akses sebagai mahasiswa dan menyimpan data ke dalam <i>sessions</i> pengguna. <p>{use case selesai}</p>
Alternative Flow	<p>A1. Menangani Identifikasi Tipe</p> <p>Pada {mengidentifikasi tipe} jika sistem mengidentifikasi tipe sebagai lembaga, maka:</p> <p>{otentikasi lembaga}</p> <ol style="list-style-type: none"> 1. Sistem melakukan otentikasi dengan basis data lembaga.

Tabel 4.17 Spesifikasi *Use Case Login* (lanjutan)

	<p>2. Sistem memberikan hak akses sebagai lembaga dan menyimpan data ke dalam sessions pengguna.</p> <p>{use case selesai}</p> <p>A2. Menangani Kegagalan Otentikasi</p> <p>Pada {otentikasi mahasiswa},{otentikasi lembaga} jika sistem mengalami kegagalan otentikasi, maka sistem menampilkan kembali <i>form</i> isian data beserta pesan kesalahan “NIM/Username dan Password tidak sesuai!”.</p> <p>{use case selesai}</p>
Post-conditions	Aktor mendapatkan hak akses dan berhasil masuk ke dalam sistem (<i>logged in</i>).

4.3.4.2 Spesifikasi *Use Case Logout*

Tabel 4.18 menjelaskan tujuan dan bagaimana pengguna dapat keluar dari sesi *logged in* sehingga pengguna tersebut nantinya akan kembali menjadi aktor *guest*.

Tabel 4.18 Spesifikasi *Use Case Logout*

Code	UC-02
Primary Actor	Mahasiswa/lembaga
Brief Description	<i>Use case logout</i> menjelaskan bagaimana aktor dapat keluar dari sesinya sebagai aktor tersebut.
Pre-conditions	Aktor dalam keadaan masuk di dalam sistem (<i>logged in</i>).
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memilih fungsi <i>logout</i>. <p>{konfirmasi logout}</p> <ol style="list-style-type: none"> 2. Sistem melakukan konfirmasi kepada aktor. 3. Sistem menghapus informasi sesi dan hak akses aktor. <p>{use case selesai}</p>
Alternative Flow	<p>A1. Mengkonfirmasi logout</p> <p>Pada {konfirmasi logout}, Jika aktor membatalkan konfirmasi atau mengkonfirmasi tidak, maka <i>logout</i> dinyatakan gagal, kemudian <i>use case</i> selesai.</p> <p>{use case selesai}</p>

Tabel 4.18 Spesifikasi *Use Case Logout* (lanjutan)

Post-conditions	Aktor dinyatakan berhasil keluar dari sistem (<i>logged out</i>).
------------------------	---

4.3.4.3 Spesifikasi *Use Case Memperbarui Identitas*

Tabel 4.19 menjelaskan tujuan dan bagaimana aktor Mahasiswa/lembaga dapat memperbarui atau harus melengkapi identitasnya yang diperlukan oleh sistem.

Tabel 4.19 Spesifikasi *Use Case Memperbarui Identitas*

Code	UC-03
Primary Actor	Mahasiswa/lembaga
Brief Description	<i>Use case</i> memperbarui identitas menjelaskan bagaimana aktor mahasiswa/lembaga dapat memberikan informasi identitas terbarunya agar dapat dihubungi oleh anggota DPM UB.
Pre-conditions	Aktor mahasiswa/lembaga dalam keadaan masuk ke dalam sistem (<i>logged in</i>).
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai setelah aktor melakukan <i>login</i>. <p>{pengecekan identitas}</p> <ol style="list-style-type: none"> 2. Sistem melakukan pengecekan terhadap kelengkapan identitas aktor yang terdapat pada basis data. 3. Sistem menyatakan identitas belum tersedia/belum lengkap dan harus dilengkapi. 4. Aktor mengirimkan data yang belum lengkap ke dalam sistem yang terdiri dari identitas diri, nomor <i>handphone</i> dan email. 5. Sistem {mengirim email} konfirmasi kepada email aktor. <p>{konfirmasi email}</p> <ol style="list-style-type: none"> 6. Aktor mengkonfirmasi dengan menekan <i>link</i> yang dikirimkan lewat email. 7. Sistem menyimpan data pembaruan identitas aktor. <p>{use case selesai}</p>
Alternative Flow	<p>A1. Pengecekan identitas aktor</p> <p>Pada {pengecekan identitas}, jika sistem menyatakan identitas aktor sudah lengkap, maka aktor dapat melakukan pembaruan identitas.</p>

Tabel 4.19 Spesifikasi *Use Case* Memperbarui Identitas (lanjutan)

	<p>{perbarui identitas}</p> <ol style="list-style-type: none"> 1. Aktor memperbarui data dan mengirimkan ke dalam sistem. 2. Sistem mengirimkan email konfirmasi kepada aktor. <p>{mengirim email}</p> <p>A2. Memperbarui Identitas</p> <p>Pada {perbarui identitas}, aktor tidak melakukan pembaruan identitas.</p> <p>{use case selesai}</p> <p>A3. Mengirim email konfirmasi kepada aktor</p> <p>Pada {mengirim email}, jika aktor tidak menerima email konfirmasi, maka aktor dapat meminta agar sistem mengirimkan ulang email konfirmasi.</p> <p>{konfirmasi email}</p> <p>A4. Mengkonfirmasi email</p> <p>Pada {konfirmasi email}, jika aktor tidak menekan <i>link</i> konfirmasi, maka sistem tidak menyimpan pembaruan identitas aktor.</p> <p>{use case selesai}</p>
Post-conditions	Sistem berhasil menyimpan informasi identitas terbaru mahasiswa/lembaga.

4.3.4.4 Spesifikasi *Use Case* Mengirimkan Aspirasi

Tabel 4.20 menjelaskan tujuan dan bagaimana aktor mahasiswa/lembaga dapat menyampaikan aspirasinya dengan cara mengirimkan secara *online* melalui sistem, sehingga aspirasi dapat diterima oleh anggota DPM UB dan diproses ke pihak-pihak yang terkait dengan isi dari aspirasi tersebut.

Tabel 4.20 Spesifikasi *Use Case* Mengirimkan Aspirasi

Code	UC-04
Primary Actor	Mahasiswa/lembaga
Brief Description	<i>Use case</i> mengirimkan aspirasi menjelaskan bagaimana aktor mahasiswa/lembaga dapat menyampaikan aspirasinya melalui sistem kepada anggota DPM UB.

Tabel 4.20 Spesifikasi *Use Case* Mengirimkan Aspirasi (lanjutan)

Pre-conditions	Aktor mahasiswa/lembaga masuk ke dalam sistem (<i>logged in</i>).
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memilih fungsi untuk mengirimkan aspirasi. 2. Sistem menampilkan <i>form</i> aspirasi yang berisi subjek, pilihan kategori (pribadi, lembaga dan global), isi aspirasi dan lampiran (<i>file</i> dalam format gambar). <p>{mengirimkan data aspirasi}</p> <ol style="list-style-type: none"> 3. Aktor mengisi data secara lengkap kemudian memerintahkan sistem untuk memproses. 4. Sistem mengambil data identitas aktor dari sesi <i>login</i>. 5. Sistem mengirimkan notifikasi <i>email</i> yang terdapat ID aspirasi. 6. Sistem menyimpan informasi data aspirasi yang dikirimkan. 7. Sistem menampilkan detail aspirasi. <p>{use case selesai}</p>
Alternative Flow	<p>A1. Mengirimkan data aspirasi</p> <p>Pada {mengirimkan data aspirasi}, jika aktor tidak mengisi <i>form</i> secara lengkap (kecuali lampiran), maka aktor tidak dapat mengirimkan aspirasi.</p> <p>{use case selesai}</p>
Post-conditions	Sistem berhasil mengirimkan ID aspirasi kepada aktor melalui email dan menyimpan data aspirasi ke dalam basis data.

4.3.4.5 Spesifikasi *Use Case* Mengelola Aspirasi

Tabel 4.21 menjelaskan tujuan dan bagaimana aktor operator dan administrator dapat mengelola terhadap data-data aspirasi yang dikirimkan oleh mahasiswa/lembaga. Pengelolaan yang dapat dilakukan terhadap data-data aspirasi meliputi: melihat daftar aspirasi, mencari aspirasi, mengklasifikasikan aspirasi, melihat detail aspirasi, memberikan tanggapan, mencetak detail aspirasi dan mengeksport daftar aspirasi kedalam format *excel*.

Tabel 4.21 Spesifikasi *Use Case* Mengelola Aspirasi

Code	UC-05
Primary Actor	Operator

Tabel 4.21 Spesifikasi *Use Case* Mengelola Aspirasi

Brief Description	<i>Use case</i> mengelola aspirasi menjelaskan bagaimana aktor dapat melakukan tindakan meliputi: melihat daftar aspirasi, mencari aspirasi, mengklasifikasikan aspirasi, melihat detail aspirasi, memberikan tanggapan, mencetak detail aspirasi dan mengeksplor daftar aspirasi kedalam format <i>excel</i> .
Pre-conditions	Aktor dalam keadaan masuk ke dalam sistem (<i>logged in</i>) dan terdapat data aspirasi yang tersimpan di dalam sistem.
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memilih fungsi untuk melihat data aspirasi pada dashboard sistem. <p>{memuat data aspirasi}</p> <ol style="list-style-type: none"> 2. Sistem melakukan proses pemuatan data aspirasi. <p>{menampilkan data aspirasi}</p> <ol style="list-style-type: none"> 3. Sistem menampilkan semua data aspirasi kedalam bentuk tabel yang pecah kedalam beberapa halaman (<i>pagination</i>). 4. Aktor melihat semua daftar data aspirasi yang telah dikirimkan oleh mahasiswa/lembaga. <p>{use case selesai}</p>
Alternative Flow	<p>A1. Menangani kegagalan pemuatan data</p> <p>Pada {memuat data aspirasi}, {memuat detail aspirasi}, jika sistem mengalami kegagalan dalam melakukan proses pemuatan data, maka sistem menampilkan pesan bahwa proses pemuatan data gagal dilakukan, kemudian selesai.</p> <p>{use case selesai}</p> <p>A2. Menampilkan daftar aspirasi</p> <p>Pada {menampilkan data aspirasi}, jika aktor melakukan <i>filtering</i> dan/atau pencarian, maka:</p> <ol style="list-style-type: none"> 1. Sistem akan menampilkan data berdasarkan <i>filter</i> atau kata kunci pencarian yang digunakan. 2. Aktor melakukan aktivitas untuk melihat detail aspirasi. 3. Sistem {memuat detail aspirasi} dan {menampilkan detail aspirasi}. 4. Sistem menampilkan <i>form</i> isian data untuk tanggapan aspirasi. <p>{use case selesai}</p> <p>A3. Mengklasifikasikan data aspirasi</p>

Tabel 4.21 Spesifikasi *Use Case* Mengelola Aspirasi (lanjutan)

	<p>Pada {menampilkan data aspirasi}, jika aktor memilih fitur klasifikasi data, maka sistem akan mengklasifikasikan data berdasarkan tujuan dari masing-masing aspirasi.</p> <p>A4. Mengekspor daftar aspirasi</p> <p>Pada {menampilkan data aspirasi}, jika aktor memilih untuk mengekspor daftar aspirasi, maka sistem akan melakukan proses ekspor kedalam format <i>excel</i> dan aktor secara otomatis mengunduh <i>file excel</i> tersebut.</p> <p>A5. Menampilkan detail aspirasi</p> <p>Pada {menampilkan detail aspirasi}, jika aktor memilih untuk memberikan tanggapan, maka akan diarahkan ke <i>use case</i> memberikan tanggapan (UC-07). Tapi, jika aktor memilih mencetak (<i>print</i>) detail aspirasi, maka sistem akan melakukan pencetakan terhadap detail aspirasi tersebut.</p>
Post-conditions	Aktor dapat melakukan pengelolaan dan melihat hasil pengelolaan data aspirasi yang telah tersimpan di dalam sistem.

4.3.4.6 Spesifikasi *Use Case* Melihat Statistik Aspirasi

Tabel 4.22 menjelaskan tujuan dan bagaimana aktor operator dan administrator dapat melihat data aspirasi ke dalam bentuk *chart* sehingga akan menampilkan hasil statistik dari aspirasi yang sudah masuk ke dalam sistem.

Tabel 4.22 Spesifikasi *Use Case* Melihat Statistik Aspirasi

Code	UC-06
Primary Actor	Operator
Brief Description	<i>Use case</i> melihat statistik aspirasi menjelaskan bagaimana aktor mendapatkan tampilan data ke dalam bentuk <i>chart</i> yang menampilkan hasil statistik dari data aspirasi.
Pre-conditions	Aktor dalam keadaan masuk ke dalam sistem (<i>logged in</i>) dan terdapat data aspirasi yang tersimpan di dalam sistem.
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memilih fungsi untuk melihat statistik aspirasi. <p>{memuat data aspirasi}</p> <ol style="list-style-type: none"> 2. Sistem melakukan proses pemuatan data aspirasi dari basis data.

Tabel 4.22 Spesifikasi Use Case Melihat Statistik Aspirasi (lanjutan)

	3. Sistem menampilkan data aspirasi kedalam beberapa bentuk <i>chart</i> yang menyajikan statistik aspirasi tiap fakultas, kategori, bulanan dan status aspirasi. {use case selesai}
Alternative Flow	A1. Menangani kegagalan pemuatan data Pada {memuat data aspirasi}, jika sistem mengalami kegagalan dalam melakukan proses pemuatan data, maka sistem menampilkan pesan bahwa proses pemuatan data gagal dilakukan, kemudian selesai. {use case selesai}
Post-conditions	Aktor dapat melihat hasil statistik dari data aspirasi yang telah dikirimkan oleh mahasiswa/lembaga.

4.3.4.7 Spesifikasi Use Case Memberikan Tanggapan

Tabel 4.23 menjelaskan tujuan dan bagaimana aktor mahasiswa/lembaga, operator dan administrator dapat saling menanggapi aspirasi yang telah disampaikan. Aktor operator dan administrator dalam kasus ini memiliki hak akses lebih dalam melakukan aktivitasnya yaitu dapat mencatat perkembangan/mengubah status perkembangan dari aspirasi tersebut.

Tabel 4.23 Spesifikasi Use Case Memberikan Tanggapan

Code	UC-07
Primary Actor	Operator
Brief Description	<i>Use case</i> memberikan tanggapan aspirasi menjelaskan bagaimana aktor dapat menanggapi terhadap aspirasi yang telah disampaikan.
Pre-conditions	Aktor dalam keadaan masuk ke dalam sistem (<i>logged in</i>) dan terdapat data aspirasi yang belum terselesaikan.
Basic Flow	{use case dimulai} 1. <i>Use case</i> dimulai ketika aktor memilih fungsi untuk melihat detail aspirasi. {identifikasi aktor} 2. Sistem menampilkan detail aspirasi, <i>form</i> perkembangan dan <i>form</i> tanggapan. 3. Aktor melakukan perubahan status perkembangan aspirasi kemudian memerintahkan sistem untuk memproses.

Tabel 4.23 Spesifikasi *Use Case* Memberikan Tanggapan (lanjutan)

	<ol style="list-style-type: none"> Aktor mengisi <i>form</i> tanggapan kemudian memerintahkan sistem untuk memproses. Sistem mengirimkan <i>email</i> pemberitahuan bahwa aspirasi mendapatkan tanggapan dan menyimpan tanggapan ke dalam basis data. Sistem menampilkan detail aspirasi beserta tanggapan terbaru. <p>{use case selesai}</p>
Alternative Flow	<p>A2. Menangani Identifikasi Aktor</p> <p>Pada {identifikasi aktor}, jika sistem mendeteksi aktor sebagai mahasiswa/lembaga, maka sistem tidak menampilkan <i>form</i> perkembangan kemudian lalu lanjut ke <i>flow</i> yang ke-4.</p>
Post-conditions	Sistem dapat menyimpan tanggapan aspirasi dan mengirimkan notifikasi pada mahasiswa/lembaga.

4.3.4.8 Spesifikasi *Use Case* Memantau Aspirasi

Tabel 4.24 menjelaskan tujuan dan bagaimana aktor dapat memantau aspirasi dengan menggunakan ID aspirasi yang telah didapatkan. Aktivitas ini akan menampilkan detail dari aspirasi sesuai dengan ID yang dimasukkan termasuk tanggapan-tanggapan yang terdapat pada aspirasi tersebut. Aktor juga dapat menggunakan *link* yang dikirimkan lewat *email* atau menggunakan *link* yang disematkan dalam *qrcode*.

Tabel 4.24 Spesifikasi *Use Case* Memantau Aspirasi

Code	UC-08
Primary Actor	<i>Visitor</i>
Brief Description	<i>Use case</i> memantau aspirasi menjelaskan bagaimana aktor dapat menampilkan detail aspirasi dan tanggapan-tanggapan yang didapatkan berdasarkan ID aspirasi yang dimasukkan.
Pre-conditions	Aktor harus memiliki ID aspirasi terlebih dahulu.
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> <i>Use case</i> dimulai ketika aktor memasukkan ID aspirasi pada <i>form</i> dan memerintahkan sistem untuk memprosesnya. Sistem memuat data aspirasi sesuai ID yang dimasukkan. <p>{mengambil data sesuai ID}</p>

Tabel 4.24 Spesifikasi *Use Case* Memantau Aspirasi (lanjutan)

	<p>3. Sistem menampilkan informasi detail aspirasi yang meliputi ID, identitas pengirim, kategori, subjek, isi aspirasi, status, lampiran dan tanggapan. {use case selesai}</p>
Alternative Flow	<p>A1. Menangani ID aspirasi yang tidak sesuai</p> <p>Pada {mengambil data sesuai ID}, jika sistem tidak mendapatkan ID yang sesuai, maka akan menampilkan pesan bahwa “Aspirasi tidak ditemukan”.</p> <p>{use case selesai}</p> <p>A2. Menangani ID aspirasi melalui <i>link</i> pada <i>qrcode</i> dan <i>email</i> {use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor melakukan <i>scanning</i> pada gambar <i>qrcode</i> yang terdapat pada hasil cetak aspirasi, hasil <i>scan</i> akan mendapatkan <i>link</i> yang didalamnya terdapat ID aspirasi yang di-<i>parsing</i> untuk diambil kode ID-nya. Aktor juga dapat mengklik <i>link</i> yang isinya sama seperti dalam <i>qrcode</i> yang dikirimkan lewat <i>email</i> notifikasi. 2. Sistem memuat data aspirasi sesuai ID pada <i>link</i>. {mengambil data sesuai ID} 3. Sistem menampilkan informasi detail aspirasi yang meliputi ID, identitas pengirim, kategori, subjek, isi aspirasi, status, lampiran dan tanggapan. {use case selesai}
Post-conditions	Sistem menampilkan detail dari aspirasi sesuai ID aspirasi.

4.3.4.9 Spesifikasi *Use Case* Mengelola Akun

Tabel 4.25 menjelaskan tujuan dan bagaimana aktor dapat menampilkan daftar akun lembaga dan operator, membuat akun baru dan juga dapat menghapusnya.

Tabel 4.25 Spesifikasi *Use Case* Mengelola Akun

Code	UC-09
Primary Actor	Administrator

Tabel 4.25 Spesifikasi *Use Case* Mengelola Akun

Brief Description	<i>Use case</i> mengelola akun lembaga dan operator menjelaskan bagaimana aktor dapat melihat daftar akun lembaga dan operator, membuat akun baru, dan menghapus akun.
Pre-conditions	Aktor dalam keadaan masuk ke dalam sistem (<i>logged in</i>) sebagai Administrator.
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memilih fungsi untuk mengelola akun. 2. Aktor {memilih tipe} akun lembaga. 3. Sistem {mengambil data} lembaga yang tersimpan di basis data. 4. Sistem menampilkan data daftar akun. 5. Aktor {memilih fungsi} membuat akun baru. 6. Sistem menampilkan <i>form</i> pendaftaran akun baru yang terdiri dari masukan berupa <i>username</i>, <i>password</i>, nama lembaga, nomor <i>handphone</i> dan <i>email</i>. 7. Aktor wajib mengisi <i>form</i> secara lengkap kemudian memerintahkan sistem untuk memproses. 8. Sistem menyimpan data akun. 9. Sistem mengirimkan <i>email</i> informasi pembuatan akun tersebut kepada <i>email</i> yang didaftarkan. <p>{use case selesai}</p>
Alternative Flow	<p>A1. Menangani Pemilihan Tipe</p> <p>Pada {memilih tipe}, jika aktor tidak memilih lembaga melainkan memilih operator, maka sistem akan mengambil data operator yang tersimpan di sistem.</p> <p>{mengambil data}</p> <p>A2. Menangani Pengambilan data</p> <p>Pada {mengambil data}, jika sistem gagal mengambil data/data tidak tersedia, maka sistem menampilkan pesan bahwa data akun lembaga tidak tersedia.</p> <p>A3. Memilih Fungsi</p> <p>Pada {memilih fungsi}, jika aktor memilih fungsi hapus akun pada akun yang dipilih, maka:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pesan konfirmasi penghapusan akun.

Tabel 4.25 Spesifikasi *Use Case* Mengelola Akun (lanjutan)

	2. Aktor melakukan {konfirmasi penghapusan} akun. 3. Data akun dihapus dari basis data. {use case selesai} A4. Menangani konfirmasi penghapusan Pada {konfirmasi penghapusan} , jika aktor tidak mengkonfirmasi, maka sistem membatalkan penghapusan. {use case selesai}
Post-conditions	Aktor dapat melihat hasil pengelolaan akun lembaga dan operator yang telah tersimpan di dalam sistem

4.3.4.10 Spesifikasi *Use Case* Menghapus Aspirasi

Tabel 4.26 menjelaskan tujuan dan bagaimana aktor administrator dapat memisahkan data aspirasi yang tidak penting kedalam tempat yang lain (tong sampah), dengan kata lain data aspirasi tersebut dihapus dari data utama. Data aspirasi ini tidak dapat dihapus secara permanen demi mengamankan data dari penyalahgunaan wewenang pemangku kepentingan dan kemungkinan kebutuhan data dikemudian hari.

Tabel 4.26 Spesifikasi *Use Case* Menghapus Aspirasi

Code	UC-10
Primary Actor	Administrator
Brief Description	<i>Use case</i> menghapus aspirasi menjelaskan bagaimana aktor administrator dapat menghapus aspirasi dari data utama dan memasukkannya ke dalam tong sampah.
Pre-conditions	Aktor dalam keadaan masuk ke dalam sistem (<i>logged in</i>) sebagai administrator dan terdapat data aspirasi yang ingin dihapus di dalam data utama.
Basic Flow	{use case dimulai} 1. <i>Use case</i> dimulai ketika aktor {memilih fungsi} untuk menghapus suatu aspirasi dari daftar aspirasi. 2. Sistem menampilkan pesan konfirmasi penghapusan aspirasi. 3. Aktor melakukan {konfirmasi penghapusan} aspirasi. 4. Data aspirasi dipindahkan ke data tong sampah {use case selesai}

Tabel 4.26 Spesifikasi *Use Case* Menghapus Aspirasi (lanjutan)

Alternative Flow	<p>A1. Menangani Pemilihan Fungsi</p> <p>Pada {memilih fungsi}, jika aktor memilih fungsi lihat tong sampah aspirasi, maka:</p> <ol style="list-style-type: none"> 1. Sistem akan menampilkan daftar aspirasi yang sudah dihapus. 2. Aktor memerintahkan sistem untuk mengembalikan data. 3. Sistem mengembalikan data ke data utama daftar aspirasi. <p>{use case selesai}</p> <p>A2. Menangani konfirmasi penghapusan</p> <p>Pada {konfirmasi penghapusan}, jika aktor tidak mengkonfirmasi, maka sistem membatalkan penghapusan.</p> <p>{use case selesai}</p>
Post-conditions	Aktor berhasil menghilangkan data aspirasi dari data daftar aspirasi utama.

BAB 5 ANALISIS KOMPONEN DAN PERANCANGAN

Bab ini menjelaskan tentang tahapan analisis komponen dan perancangan Sistem Informasi Aspirasi *Online* berbasis web pada DPM UB. Berdasarkan pada diagram pemodelan *reuse-oriented development* yang menerapkan pengembangan berorientasi penggunaan ulang, maka tahapan awal sebelum melakukan perancangan adalah melakukan analisis komponen dan melakukan modifikasi persyaratan terhadap komponen *reuse* tersebut agar sesuai dengan spesifikasi persyaratan yang telah ditentukan, setelah komponen sesuai dengan persyaratan tahapan berikutnya adalah melakukan perancangan menggunakan komponen *reuse* yang telah didapatkan untuk merancang suatu sistem baru.

Menurut Tahir, et al., (2016) kerangka kerja rekayasa perangkat lunak berbasis komponen dilakukan dengan tiga tahapan, yaitu: *Component selection* yang dilakukan pada tahapan analisis dan perancangan, *interface with component* dilakukan pada saat implementasi dan pengujian, dan *replacement* dilakukan pada saat pemeliharaan sistem.

5.1 Analisis Komponen

Tahap analisis komponen dilakukan untuk mencari komponen yang sesuai dengan spesifikasi persyaratan (Sommerville, 2011). Berikut ini merupakan hasil pemilihan komponen yang dapat digunakan ulang dalam pengembangan sistem informasi aspirasi *online* pada DPM UB. Sommerville dalam bukunya menambahkan bahwa komponen yang telah ditentukan biasanya tidak ada kecocokan yang tepat dan komponen yang dapat digunakan hanya tersedia beberapa fungsi saja. Sehingga diperlukan perubahan atau penambahan algoritma program saat mengimplementasikannya pada pengembangan sistem, masing-masing komponen *reuse* atau bagian dari komponen akan diintegrasikan untuk membuat sistem baru. Penjelasan dari komponen-komponen yang akan digunakan ulang dan tujuan penggunaannya diuraikan pada Tabel 5.1 berikut ini.

Tabel 5.1 Daftar komponen yang dapat digunakan ulang

No	Nama Komponen	Deskripsi	Kegunaan
1	CodeIgniter	Merupakan <i>Back-end Framework</i> yang dapat digunakan dalam mengimplementasikan keseluruhan sistem. Sumber: https://codeigniter.com	Digunakan sebagai kerangka dasar <i>back-end</i> sistem.
2	Source code LAPOR BEM	Kode program dari fungsi-fungsi yang disediakan sistem LAPOR BEM (Layanan Aspirasi dan Pengaduan <i>Online</i> Terhadap	Digunakan untuk memudahkan

Tabel 5.1 Daftar komponen yang dapat digunakan ulang

No	Nama Komponen	Deskripsi	Kegunaan
		<p>BEM FILKOM UB) meliputi logika pengolahan data yang terdapat pada kelas-kelas di dalam <i>controller</i> dan <i>model</i> dari sistem tersebut, diambil untuk digunakan kembali pada sistem yang akan dikembangkan.</p> <p>Sumber: http://bem.filkom.ub.ac.id/apps/lapor</p>	dalam pembuatan fungsi dan logika yang dibutuhkan sistem.
3	Web Service API BAIS	<p>Layanan otentikasi akun mahasiswa yang disediakan oleh Brawijaya Authentication and Identification System (BAIS) untuk membuktikan keaslian identitas diri yang digunakan oleh mahasiswa.</p> <p>Sumber: https://bais.ub.ac.id/api</p>	API ini digunakan sebagai otentikasi mahasiswa yang akan melakukan <i>login</i> pada sistem.
4	PHPMailer dan SMTP	<p>Suatu pustaka kode yang dapat mengirimkan email dengan aman dan mudah melalui kode PHP sederhana dari <i>server web</i> melalui protokol SMTP.</p> <p>Sumber: https://github.com/PHPMailer</p>	Komponen ini digunakan untuk mengirimkan notifikasi <i>email</i> dari sistem kepada <i>user</i>
5	Ci PHP Qrcode	<p>PHP QR Code adalah open source (LGPL) library untuk menghasilkan QR Code. Berdasarkan “libqrencode C library”, menyediakan API untuk membuat gambar Kode QR (PNG/JPEG).</p> <p>Sumber: http://phpqrcode.sourceforge.net</p>	Digunakan untuk membuat gambar QRCode yang dapat di scan oleh pengguna untuk melihat detail aspirasi.
6	Time ago Helpers	<p>Helpers yang dapat digunakan CI untuk mengubah format tanggal seperti format tanggal <i>facebook</i> dan <i>twitter</i>, contoh: “3 Menit yang lalu”, “1 jam yang lalu”.</p> <p>Sumber: https://bismalabs.co.id</p>	Digunakan untuk menampilkan waktu tanggapan dikirimkan,

Tabel 5.1 Daftar komponen yang dapat digunakan ulang (lanjutan)

No	Nama Komponen	Deskripsi	Kegunaan
			sesuai format tersebut.
7	Mobile_Detect	Mobile_Detect adalah pustaka kelas PHP yang ringan untuk mendeteksi perangkat seluler (termasuk tablet). Ini menggunakan string User-Agent yang dikombinasikan dengan header HTTP tertentu untuk mendeteksi lingkungan mobile. Sumber: http://mobiledetect.net	Digunakan untuk mendeteksi dan memaksimalkan tampilan sistem pada browser mobile dan tablet.
8	AdminLTE-3.0	Kerangka tampilan dashboard admin yang open source. Dibangun menggunakan dasar Bootstrap 3, AdminLTE menyediakan berbagai komponen yang responsif, dapat digunakan kembali, dan sudah umum digunakan. Sumber: https://adminlte.io	Digunakan sebagai kerangka dasar keseluruhan <i>front-end</i> sistem.

5.2 Modifikasi Persyaratan

Tahapan modifikasi persyaratan (*requirements modification*) merupakan tahapan lanjutan dalam menganalisis komponen, tahapan ini menganalisis modifikasi yang harus dilakukan agar sesuai, komponen *reuse* sudah memenuhi persyaratan namun membutuhkan sedikit modifikasi agar sesuai. Tabel 5.2 berikut ini merupakan penjelasan modifikasi yang akan dilakukan pada tahapan implementasi dan integrasi dari beberapa komponen yang belum sepenuhnya sesuai dengan spesifikasi persyaratan sistem.

Tabel 5.2 Daftar Modifikasi Komponen

No	Nama	Sebelum	Sesudah
1	Web Service API BAIS	<i>Output</i> yang didapatkan dari API BAIS berupa JSON.	Diubah kedalam format array dan diambil beberapa <i>value</i> saja.
2	Time ago Helpers	Menggunakan bahasa inggris, contoh: "3 minutes ago".	Menjadi bahasa indonesia, contoh: "3 menit yang lalu".

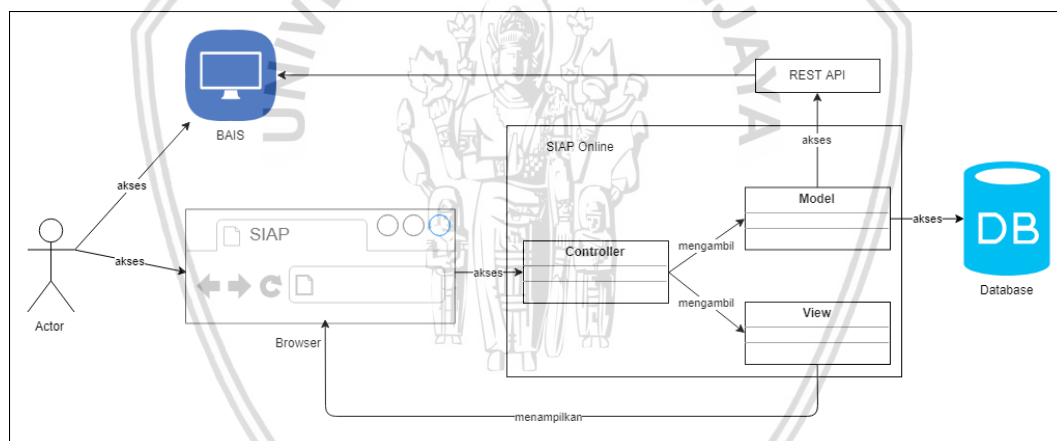
5.3 Perancangan dengan Komponen

Perancangan dibangun berdasarkan hasil studi literatur, analisis persyaratan sistem, analisis komponen dan modifikasi yang dilakukan. Sehingga pada tahapan perancangan dengan komponen juga akan menggunakan apa yang dapat digunakan dari komponen *reuse* yang sudah dianalisis untuk digunakan ulang.

Tahap perancangan meliputi perancangan arsitektur sistem, perancangan interaksi objek, perancangan kelas objek, perancangan basis data, perancangan komponen baru atau melakukan integrasi terhadap komponen *reuse* yang sudah didapatkan sebelumnya, dan melakukan perancangan desain antarmuka pengguna.

5.3.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem digunakan untuk mengidentifikasi keseluruhan struktur sistem, komponen utama, relasi dan bagaimana pendistribusiannya (Sommerville, 2011). Secara tidak langsung perancangan arsitektur sistem dilakukan untuk menggambarkan jalannya sistem mulai dari proses pengguna mengakses sistem, pemrosesan data hingga menghasilkan *output* bagi pengguna.



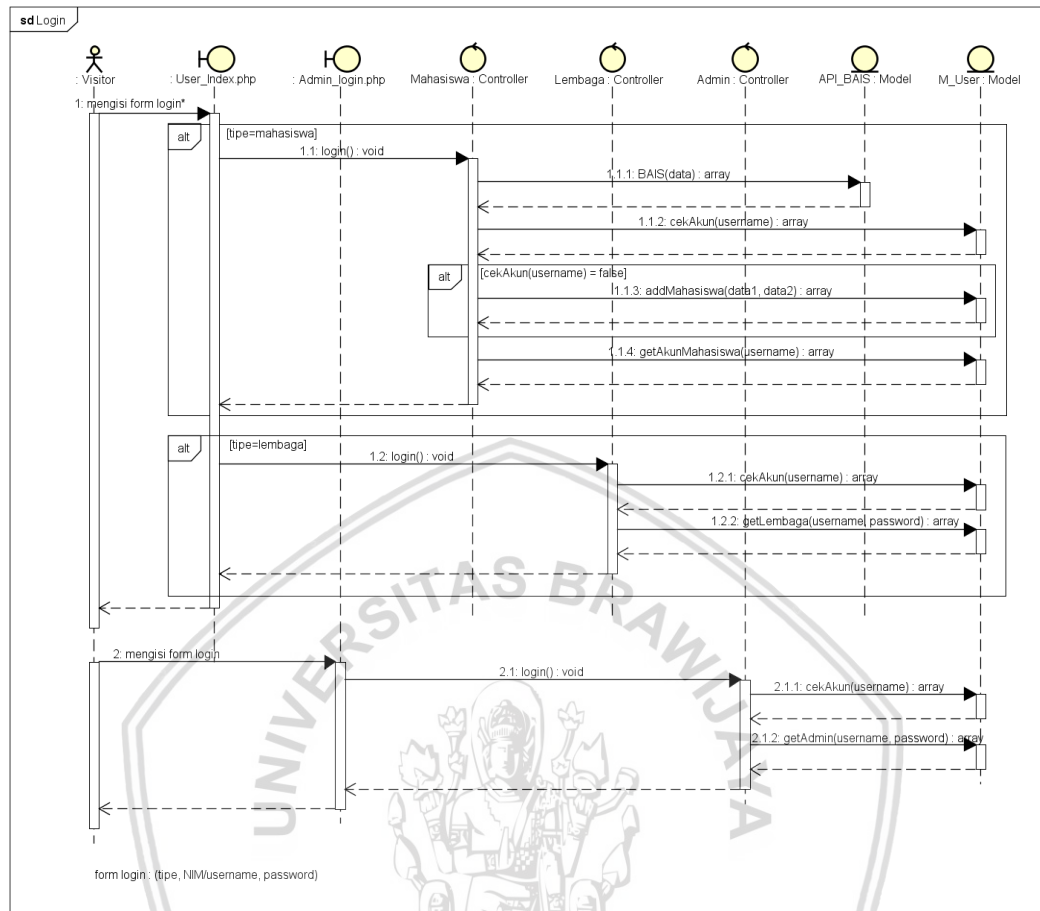
Gambar 5.1 Rancangan Arsitektur Sistem

Pada Gambar 5.1 menjelaskan bahwa sistem diakses oleh pengguna melalui web *browser*. Sistem dibangun dan dikembangkan dengan menerapkan pola perancangan MVC (*Model-View-Controller*). Sistem menggunakan *web service* REST API untuk pertukaran data dengan sistem lain.

5.3.2 Perancangan Interaksi Objek

Interaksi objek (*sequence*) menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. Pada tahap ini dari keseluruhan interaksi objek dalam sistem, hanya merancang lima sampel diagram interaksi saja. Berikut ini merupakan rancangan diagram interaksi dari *SIAP Online* DPM UB.

5.3.2.1 Diagram Interaksi Login



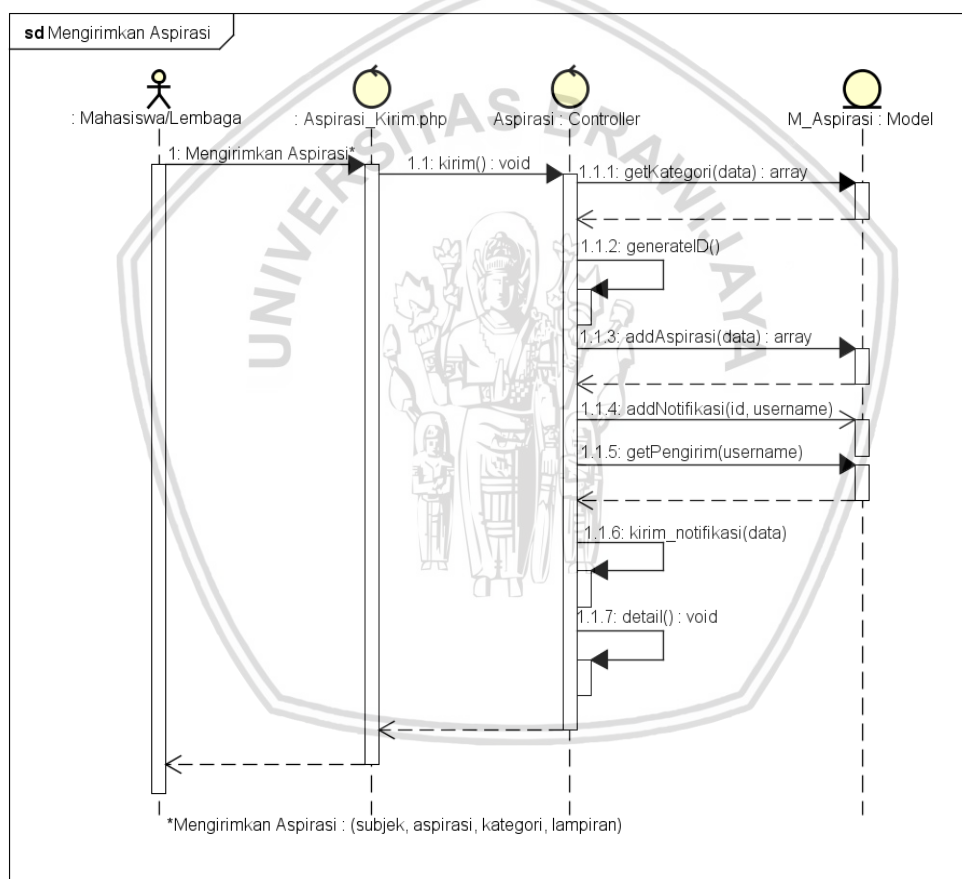
Gambar 5.2 Diagram Interaksi Login

Pada gambar 5.2 merupakan diagram interaksi *login* yang dimulai ketika aktor mengakses halaman index dan mengisi *form login*, kemudian aktor memilih tipe sebagai mahasiswa atau lembaga, sedangkan untuk halaman login admin terdapat pada halaman yang berbeda. Jika tipe yang dipilih adalah sebagai mahasiswa maka sistem meminta masukan berupa NIM dan kata sandi, sedangkan untuk lembaga menggunakan *username* dan kata sandi. Aktor yang login sebagai mahasiswa maka akan diarahkan ke fungsi login yang terdapat pada *controller* mahasiswa. Controller mahasiswa akan melakukan *request* pada API_BAIS dengan menggunakan fungsi BAIS yang berisi parameter data array, dalam model API_BAIS ini NIM dan kata sandi yang dikirimkan akan dilakukan otentikasi menggunakan *web service* yang disediakan oleh *bais.ub.ac.id/api*. Jika kombinasi NIM dan kata sandi sesuai maka akan mendapatkan balasan berupa identitas dari nim tersebut, namun jika tidak sesuai maka akan mendapatkan balasan pesan kesalahan. Dari data balasan tersebut fungsi BAIS akan mengembalikan data dalam bentuk array. Setelah itu, controller mahasiswa akan melakukan pengecekan NIM pada database melalui fungsi *cekAkun* dalam model M_User. Jika *username*/NIM mahasiswa belum tersimpan dalam database maka akan melakukan penyimpanan melalui fungsi *addMahasiswa*. Jika data sudah tersimpan

maka melakukan pengambilan data mahasiswa dari database melalui fungsi `getAkunMahasiswa` dan dimasukkan ke dalam *session* sehingga status aktor tersebut akan berubah menjadi dalam keadaan login.

Sedikit berbeda untuk tipe lembaga, *controller* yang digunakan adalah *controller* Lembaga dan langsung melakukan `cekAkun` tanpa melalui `API_BAIS` terlebih dulu dan jika berhasil maka akan mengambil data dari database melalui fungsi `getAkun` yang juga kembalinya dimasukkan ke dalam *session*. Sedangkan untuk aktor yang ingin masuk sebagai admin harus menggunakan halaman yang berbeda yaitu `Admin_Login.php` yang mengirimkan isi dari *form login* ke *controller* Admin yang juga melakukan `cekAkun` ke model `M_User`, untuk mengambil data menggunakan fungsi `getAdmin`.

5.3.2.2 Diagram Interaksi Mengirimkan Aspirasi

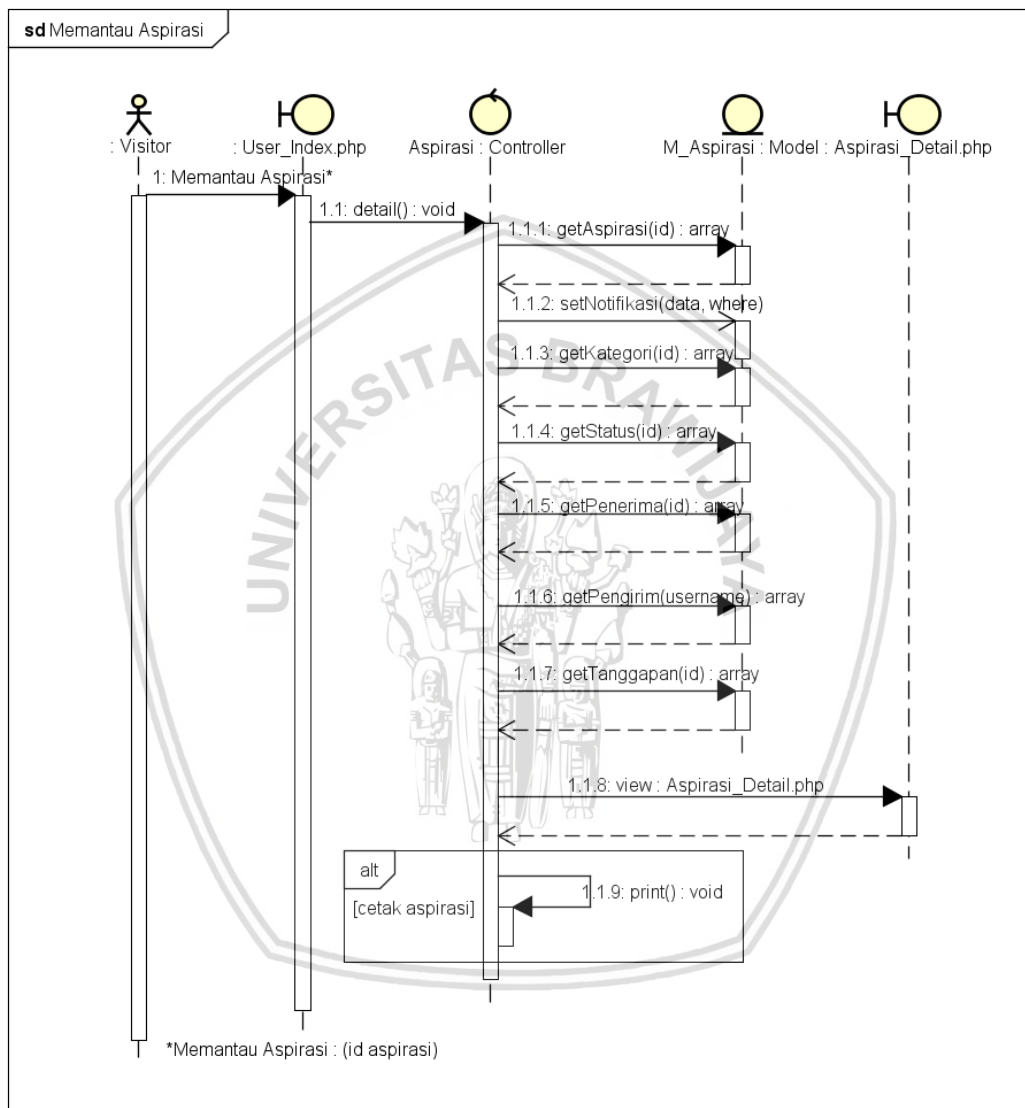


Gambar 5.3 Diagram Interaksi Mengirimkan Aspirasi

Pada Gambar 5.3 merupakan diagram interaksi mengirimkan aspirasi yang dimulai dari aktor mahasiswa/lembaga memilih fungsi `kirim()`, fungsi tersebut jika tidak menerima *method* POST maka akan menampilkan *form* Aspirasi yang terdiri dari kolom Subjek, pilihan kategori yang di dapat dari `getKategori()`, kolom konten aspirasi dan kolom lampiran yang menerima masukan berupa gambar. Ketika *form* tersebut di *submit* maka akan melakukan penyimpanan data aspirasi melalui fungsi `addAspirasi()` namun sebelum melakukan itu diperlukan `generateID()` untuk

membuat ID aspirasi. Jika aspirasi sukses disimpan maka akan memberikan notifikasi pada admin/operator sistem melalui fungsi addNotifikasi() dan mengirimkan email notifikasi kepada pengirim yang berisi ID Aspirasi menggunakan fungsi kirim_notifikasi(data), lalu sistem akan mengarahkan ke fungsi detail() yang juga terdapat dalam kelas *controller* Aspirasi untuk menampilkan detail aspirasi.

5.3.2.3 Diagram Interaksi Memantau Aspirasi

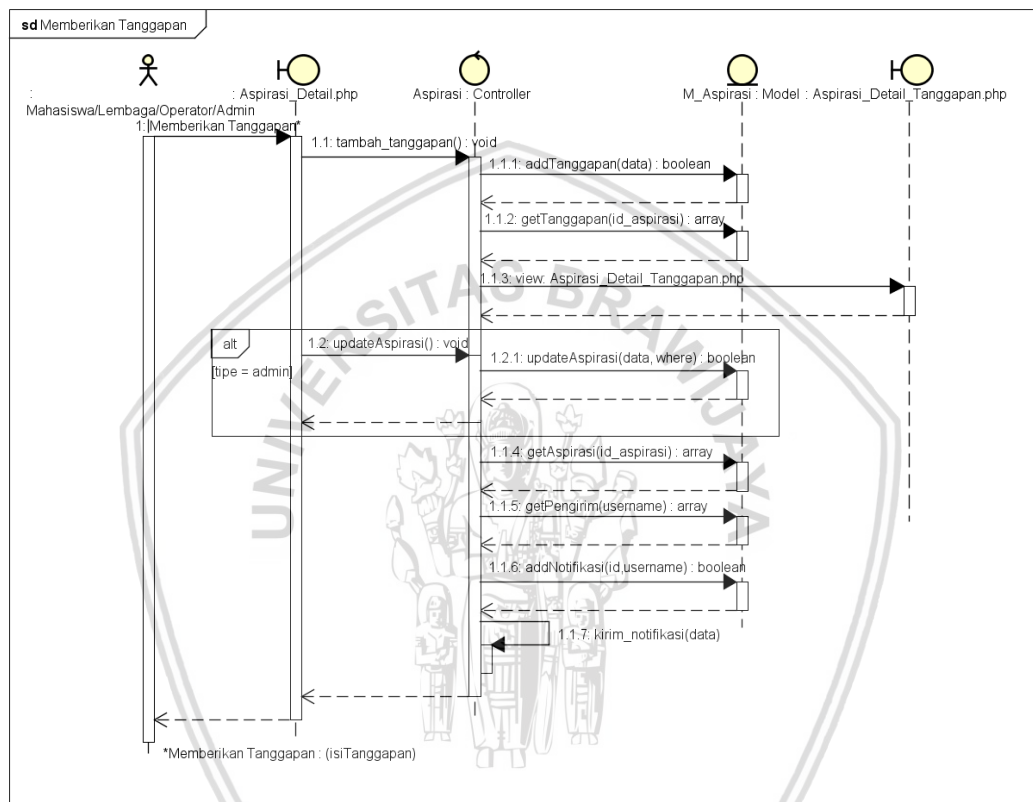


Gambar 5.4 Diagram Interaksi Memantau Aspirasi

Pada Gambar 5.4 merupakan diagram interaksi memantau aspirasi, aktor tidak perlu melakukan login, siapapun yang mempunyai ID aspirasi dapat memantau langsung melalui sistem tanpa perlu *login* terlebih dahulu. Sistem menyediakan *form tracking* untuk memantau aspirasi di halaman awal sistem, selanjutnya aktor dapat memasukkan ID aspirasi yang dimilikinya lalu dikirimkan ke dalam sistem melalui fungsi detail() yang terdapat pada *controller* Aspirasi, sistem akan mengambil data aspirasi sesuai ID yang dimasukkan menggunakan fungsi getAspirasi() yang disediakan oleh model M_Aspirasi, jika ID ditemukan

maka sistem melakukan *update* notifikasi bahwa aspirasi tersebut telah dibaca, lalu melakukan pengambilan data ketegori, status, penerima, pengirim, dan tanggapan-tanggapan aspirasi sesuai ID aspirasi tersebut. Selanjutnya menampilkan detail aspirasi tersebut menggunakan *view* Aspirasi_Detail.php dan aktor dapat mencetak detail aspirasi tersebut melalui fungsi *print()*. Namun, jika ID tidak ditemukan didalam database maka aktor akan mendapatkan pesan bahwa aspirasi tidak ditemukan.

5.3.2.4 Diagram Interaksi Memberikan Tanggapan



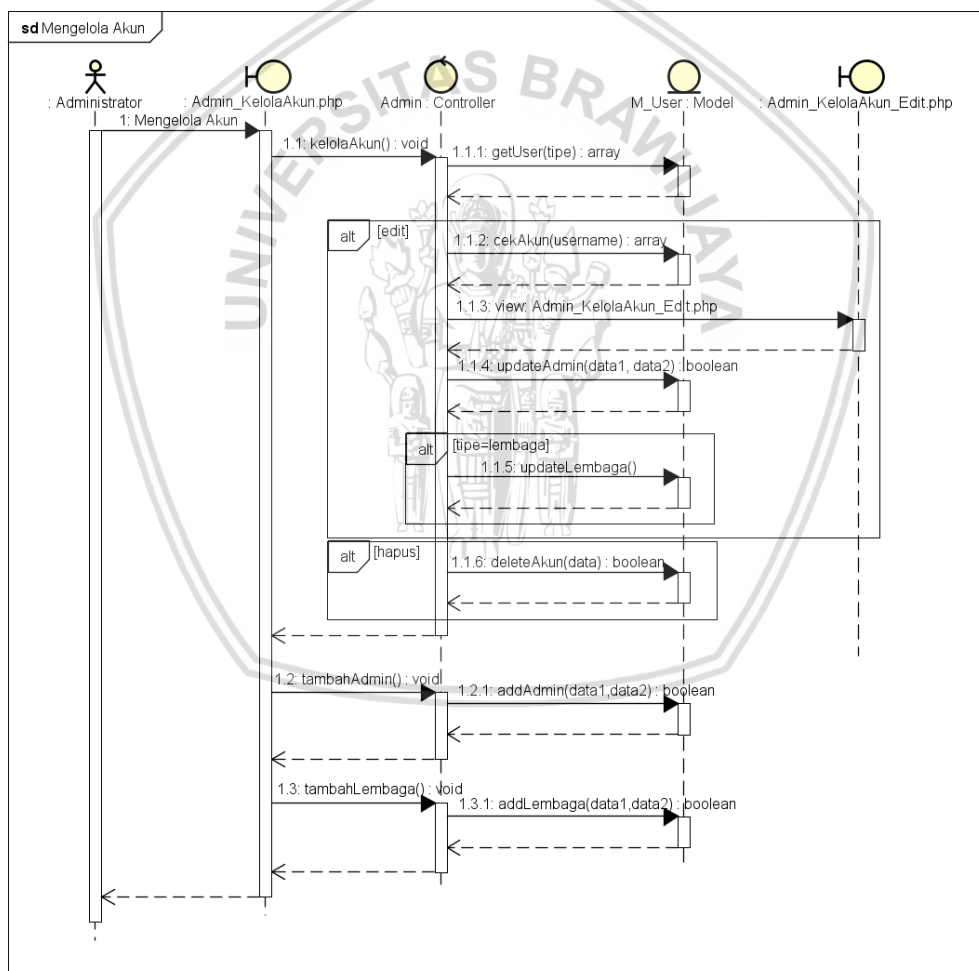
Gambar 5.5 Diagram Interaksi Memberikan Tanggapan

Pada Gambar 5.5 semua aktor yang telah *login* sistem dapat memberikan tanggapannya melalui *form* tanggapan yang terdapat pada halaman detail aspirasi, tanggapan dikirimkan melalui fungsi *tambah_tanggapan()* pada *controller* Aspirasi dan melakukan penyimpanan data melalui fungsi *addTanggapan()* yang terdapat pada model *M_Aspirasi*, jika tanggapan sukses disimpan selanjutnya mengambil semua tanggapan menggunakan *getTanggapan()* lalu ditampilkan menggunakan *view* *Aspirasi_Detail_Tanggapan.php*. Jika Aktor sebagai admin (operator/administrator) maka dapat melakukan *update* status melalui fungsi *updateAspirasi()*. Selanjutnya *controller* mengambil data aspirasi dan pengirim untuk menambahkan notifikasi termasuk mengirimkan email notifikasi.

5.3.2.5 Diagram Interaksi Mengelola Akun

Pada Gambar 5.6 diagram interaksi mengelola akun hanya bisa dilakukan oleh aktor administrator, dalam interaksi ini aktor dapat melihat *list* akun yang

terdaftar, mengedit, menghapus akun yang sudah dibuat sebelumnya dan membuat akun baru untuk aktor bertipe admin dan lembaga. Prosesnya dimulai dari aktor yang masuk ke fungsi `kelolaAkun()` yang terdapat dalam *controller* Admin dan aktor memilih tipe akun yang akan dikelola. Tipe akun yang dimaksud adalah tipe admin (termasuk operator), lembaga dan mahasiswa. Setelah memilih tipe, sistem mengambil semua data akun berdasarkan tipe tersebut lalu aktor dapat melakukan edit, hapus atau membuat akun baru. Pada aktivitas edit dan hapus akun, sistem melakukan penghapusan terhadap data akun yang dipilih, sedangkan untuk membuat akun baru sistem akan menampilkan *form* pendaftaran akun baru sesuai tipe yang dipilih lalu aktor dapat mengisi *form* tersebut dan mengirimkan datanya ke sistem. Sistem akan menyimpan data pembuatan akun setelah itu akun dapat digunakan untuk *login*. Fungsi Edit, hapus dan tambah tidak berlaku untuk akun yang bertipe mahasiswa, karena otentikasinya menggunakan *web service* dari BAIS.



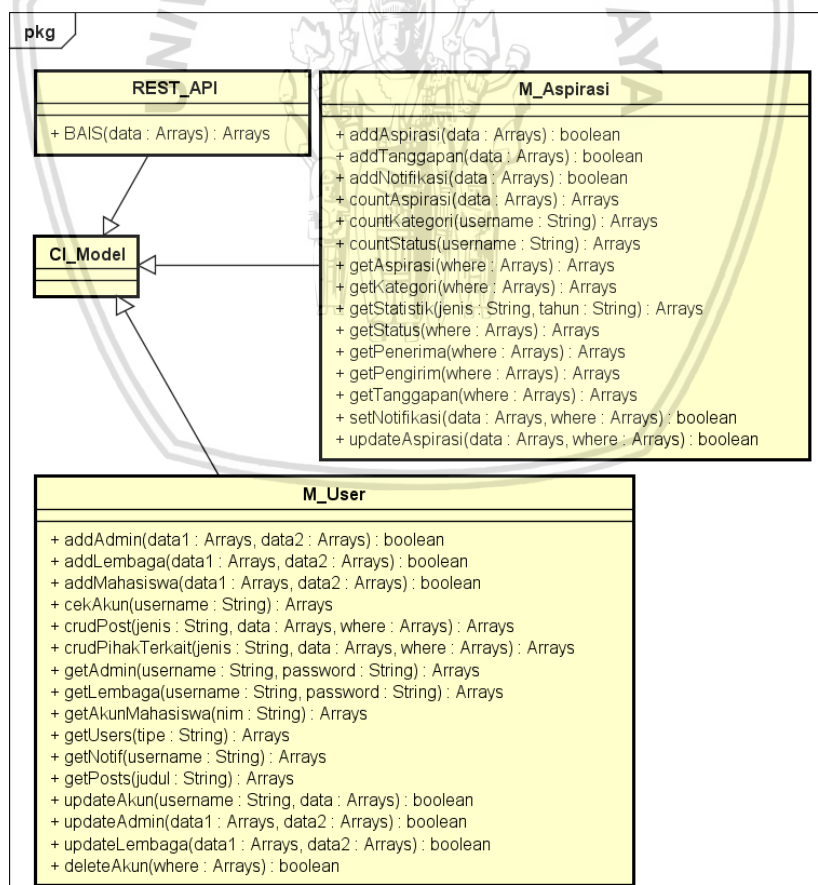
Gambar 5.6 Diagram Interaksi Mengelola Akun

5.3.3 Perancangan Kelas Objek

Kumpulan dari kelas objek akan digambarkan kedalam suatu kelas diagram, menurut Sommerville (2011) kelas diagram merupakan diagram yang menunjukkan kelas objek dalam sistem dan asosiasi diantara kelas-kelas tersebut. Dalam tahap pembangunan sistem akan menggunakan pola perancangan MVC (*Model, View, Controller*), sehingga setiap kelas yang didefinisikan dibagi ke dalam beberapa kelas *model* dan kelas *controller*.

5.3.3.1 Diagram Kelas Model

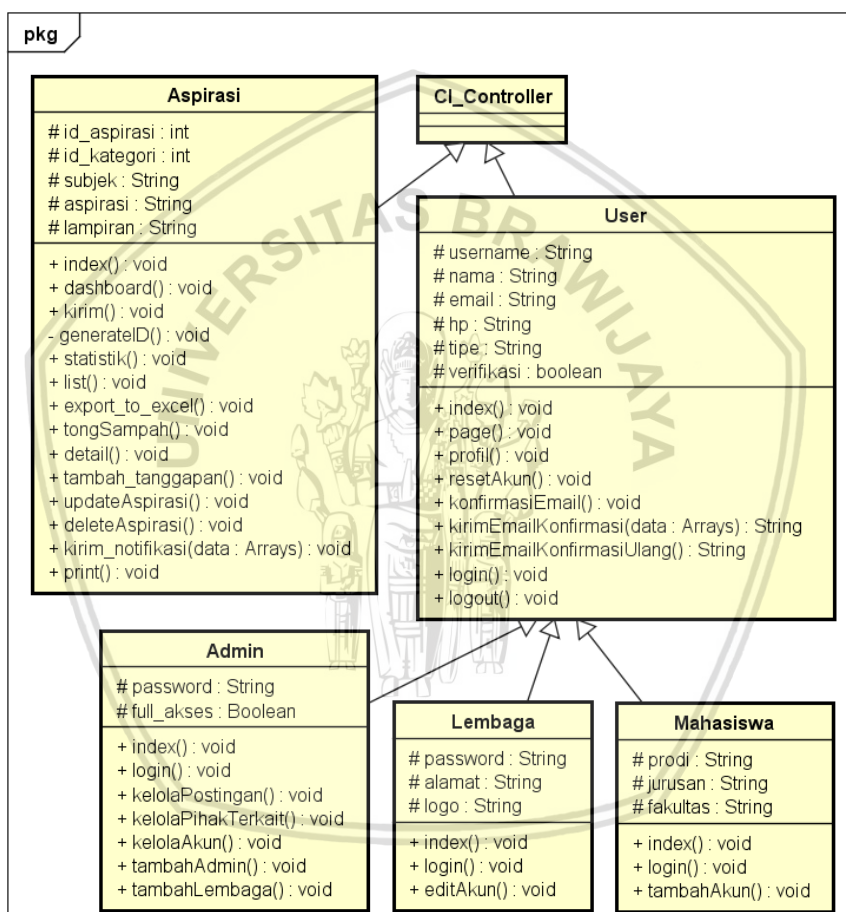
Kelas *model* merupakan kelas-kelas hasil generalisasi atau kelas yang mewarisi fungsi dari kelas *CI_Model*. Kelas *model* berfungsi untuk operasi data pada *database*, baik itu untuk operasi memasukan data, *update* data, mengambil data, atau pun menghapus data. Kelas *model* terdiri dari kelas *M_User* dan kelas *M_Aspirasi*. Kelas *M_Model* berfungsi untuk operasi data akun, baik akun mahasiswa, lembaga ataupun akun administrator dan operator. Kelas *M_Aspirasi* berfungsi untuk operasi data yang berkaitan dengan data aspirasi. Sedangkan Kelas *REST_API* merupakan sebuah kelas yang digunakan untuk mengirim dan menerima *web service* dari BAIS (Brawijaya Authentication and Identification System). Kelas-kelas Model digambarkan pada pada Gambar 5.7 berikut ini.



Gambar 5.7 Diagram Kelas Model

5.3.3.2 Diagram Kelas Controller

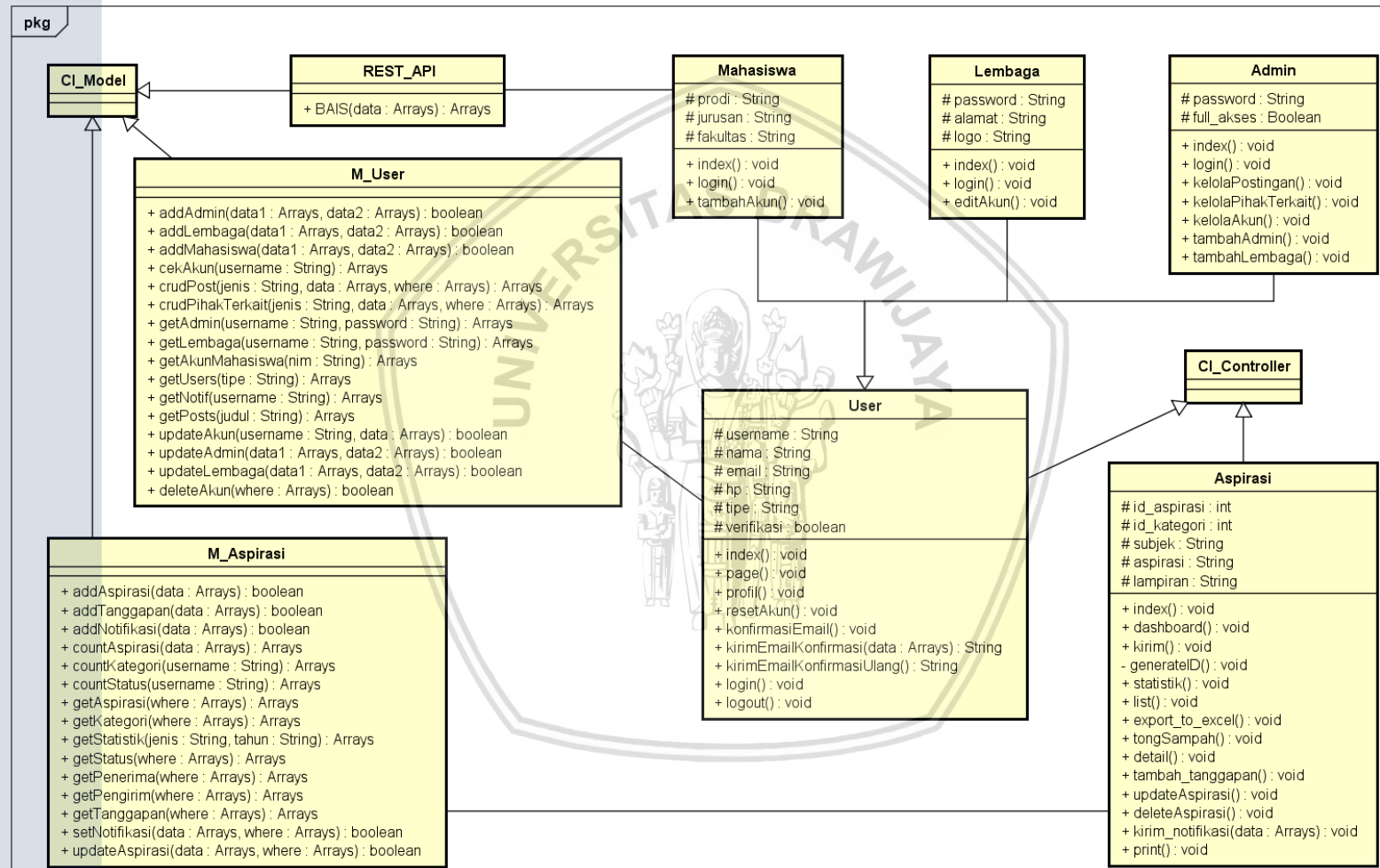
Kelas *controller* merupakan kelas-kelas hasil generalisasi atau kelas yang mewarisi fungsi dari kelas *CI_Controller*. Kelas *controller* tersebut terdiri dari kelas Aspirasi, User, Admin, Lembaga, dan Mahasiswa. Untuk kelas Admin, Lembaga dan Mahasiswa merupakan kelas turunan dari kelas User. Kelas Aspirasi secara umum berfungsi untuk pemrosesan aktivitas yang berkaitan dengan pengelolaan aspirasi. Sedangkan kelas User berfungsi untuk memberikan akses kepada setiap aktor berdasarkan jenis akun. Kelas-kelas turunan User menangani semua aktivitas yang berkaitan dengan akun user. Desain diagram digambarkan pada Gambar 5.8 berikut ini.



Gambar 5.8 Diagram Kelas Controller

5.3.3.3 Diagram Relasi Kelas Model dengan Controller

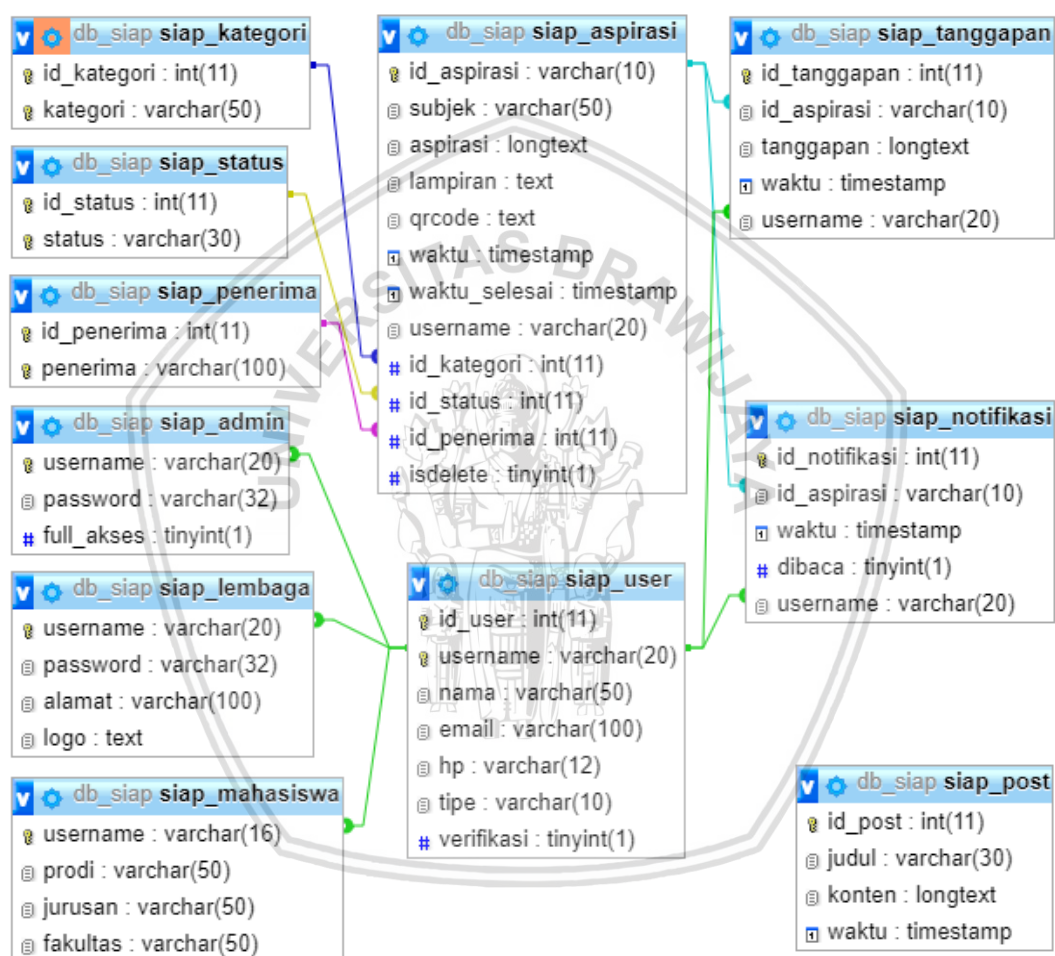
Hasil desain relasi perancangan kelas dari sistem informasi aspirasi *online* (SIAP *Online*) pada DPM UB yang akan dibangun pada tahapan implementasi terdapat pada Gambar 5.9. Sistem ini dibangun menggunakan *framework* Codeigniter sehingga semua kelas yang akan dibuat adalah hasil generalisasi dari kelas *CI_Controller* dan *CI_Model*, kedua kelas tersebut merupakan *parent* dari kelas *controller* dan kelas *model* yang telah disediakan oleh CodeIgniter.



Gambar 5.9 Diagram Relasi Kelas Model dengan Controller

5.3.4 Perancangan Basis Data

Rancangan basis data pada Sistem Informasi Aspirasi *Online* DPM UB terdapat pada Gambar 5.10, rancangan ini dibuat berdasarkan pemodelan kelas diagram pada tahapan sebelumnya yang ditunjukkan dalam Gambar 5.9. Beberapa kelas pada kelas diagram direpresentasikan oleh tabel yang terdapat pada PDM (*physical data model*), sedangkan atribut yang terdapat di dalam masing-masing kelas direpresentasikan oleh nama-nama kolom pada setiap tabel di PDM. Hubungan antar tabel pada PDM dibuat berdasarkan hubungan antar kelas pada kelas diagram.



Gambar 5.10 Perancangan Basis Data

Terdapat sebelas tabel pada basis data Sistem Informasi Aspirasi *Online* pada DPM UB yang terdiri dari tabel siap_aspirasi, siap_tanggapan, siap_notifikasi, siap_user, siap_mahasiswa, siap_lembaga, siap_admin, siap_penerima, siap_status, siap_kategori dan siap_post. Berikut ini merupakan struktur dan atribut yang terdapat pada masing-masing tabel serta keterangannya.

Tabel 5.3 Struktur Tabel siap_aspirasi

Nama Atribut	Tipe	Lebar	Keterangan
id_aspirasi	varchar	20	sebagai <i>primary key</i>
subjek	varchar	50	menyimpan subjek aspirasi
aspirasi	longtext		menyimpan isi/konten aspirasi
lampiran	text		menyimpan <i>link</i> lampiran
qrcode	text		menyimpan <i>link</i> qrcode
waktu	timestamp		bawaan = <i>current_timestamp</i>
waktu_selesai	timestamp		menyimpan waktu aspirasi selesai
username	varchar	20	menyimpan <i>username</i> pengirim
id_kategori	int	11	<i>foreign key</i> dari tabel siap_kategori
id_status	int	11	<i>foreign key</i> dari tabel siap_status
id_penerima	int	11	<i>foreign key</i> dari tabel siap_penerima
isdelete	tinyint	1	boolean untuk menandakan aspirasi dihapus/tidak

Pada Tabel 5.3 di atas menunjukkan struktur tabel siap_aspirasi. Tabel tersebut digunakan untuk menyimpan data terkait aspirasi yang disampaikan mahasiswa maupun lembaga.

Tabel 5.4 Struktur Tabel siap_tanggapan

Nama Atribut	Tipe	Lebar	Keterangan
id_tanggapan	int	11	sebagai <i>primary key</i> dan <i>auto increment</i>
id_aspirasi	varchar	10	<i>foreign key</i> dari tabel siap_aspirasi
tanggapan	longtext		menyimpan isi/konten tanggapan
waktu	timestamp		bawaan = <i>current_timestamp</i>
username	varchar	20	<i>foreign key</i> dari tabel siap_user

Pada Tabel 5.4 di atas menunjukkan struktur tabel siap_tanggapan. Tabel tersebut digunakan untuk menyimpan data terkait tanggapan-tanggapan yang didapatkan pada masing-masing aspirasi.

Tabel 5.5 Struktur Tabel siap_notifikasi

Nama Atribut	Tipe	Lebar	Keterangan
id_notifikasi	int	11	sebagai <i>primary key</i> dan <i>auto increment</i>
id_aspirasi	varchar	10	<i>foreign key</i> dari tabel siap_aspirasi
waktu	timestamp		bawaan = <i>current_timestamp</i>
dibaca	tinyint	1	boolean untuk menandakan aspirasi sudah dibaca/belum
username	varchar	20	<i>foreign key</i> dari tabel siap_user

Pada Tabel 5.5 menunjukkan struktur tabel siap_notifikasi. Tabel tersebut digunakan untuk menyimpan data notifikasi aspirasi pada pengguna sistem, setiap pengguna akan mendapatkan notifikasi jika terdapat tanggapan baru yang

datanya diambil dari tabel tersebut, untuk pengguna bertipe admin (operator/administrator) tabel siap_notifikasi juga digunakan untuk menampung notifikasi jika terdapat aspirasi yang baru disampaikan.

Tabel 5.6 Struktur Tabel siap_user

Nama Atribut	Tipe	Lebar	Keterangan
id_user	int	11	sebagai <i>primary key</i> & <i>auto increment</i>
username	varchar	20	identitas yang bersifat unik dari tiap pengguna
nama	varchar	50	menyimpan nama pengguna
email	varchar	100	menyimpan <i>email</i> pengguna
hp	varchar	12	menyimpan nomor hp pengguna
tipe	varchar	10	menyimpan tipe pengguna
verifikasi	tinyint	1	boolean yang menandakan <i>email</i> sudah diverifikasi/belum

Pada Tabel 5.6 di atas menunjukkan struktur tabel siap_user. Tabel tersebut merupakan tabel utama dari data pengguna, data sisanya disimpan ditabel sesuai tipe masing-masing pengguna.

Tabel 5.7 Struktur Tabel siap_mahasiswa

Nama Atribut	Tipe	Lebar	Keterangan
username	varchar	20	<i>foreign key</i> dari tabel siap_user
prodi	varchar	50	menyimpan program studi mahasiswa
jurusan	varchar	50	menyimpan jurusan mahasiswa
fakultas	varchar	50	menyimpan fakultas mahasiswa

Pada Tabel 5.7 di atas menunjukkan struktur tabel siap_mahasiswa. Tabel tersebut merupakan tabel keterangan dari tabel siap_user yang dapat menampung data program studi, jurusan dan fakultas dari mahasiswa yang menggunakan sistem.

Tabel 5.8 Struktur Tabel siap_lembaga

Nama Atribut	Tipe	Lebar	Keterangan
username	varchar	20	<i>foreign key</i> dari tabel siap_user
password	varchar	32	menyimpan md5 <i>password</i>
alamat	varchar	100	menyimpan alamat lembaga
logo	text		menyimpan <i>link</i> logo lembaga

Pada Tabel 5.8 menunjukkan struktur tabel siap_lembaga. Tabel tersebut merupakan tabel keterangan dari tabel siap_user yang dapat menampung data *password*, alamat dan logo lembaga.

Tabel 5.9 Struktur Tabel siap_admin

Nama Atribut	Tipe	Lebar	Keterangan
username	varchar	20	<i>foreign key</i> dari tabel siap_user

Tabel 5.9 Struktur Tabel siap_admin (lanjutan)

Nama Atribut	Tipe	Lebar	Keterangan
password	varchar	32	manyimpan md5 <i>password</i>
full_akses	tinyint	1	boolean yang membedakan hak akses sebagai administrator/operator sistem

Pada Tabel 5.9 di atas menunjukkan struktur tabel siap_admin. Tabel tersebut merupakan tabel keterangan dari tabel siap_user yang dapat menampung data *password* dan hak akses dari pengguna dengan tipe admin.

Tabel 5.10 Struktur Tabel siap_penerima

Nama Atribut	Tipe	Lebar	Keterangan
id_penerima	int	11	sebagai <i>primary key</i> dan <i>auto increment</i>
penerima	varchar	100	menyimpan nama penerima/pihak terkait dengan penyelesaian aspirasi

Pada Tabel 5.10 di atas menunjukkan struktur tabel siap_penerima. Tabel tersebut merupakan tabel yang menampung data pihak-pihak yang terkait dengan aspirasi yang disampaikan, pihak terkait tersebut merupakan tujuan/penerima dari aspirasi yang telah disampaikan.

Tabel 5.11 Struktur Tabel siap_status

Nama Atribut	Tipe	Lebar	Keterangan
id_status	int	11	sebagai <i>primary key</i> dan <i>auto increment</i>
status	varchar	30	menyimpan nama status

Pada Tabel 5.11 di atas menunjukkan struktur tabel siap_status. Tabel tersebut merupakan tabel yang menampung data jenis-jenis nama dari status aspirasi, ini digunakan untuk menandai/mencatat perkembangan dari aspirasi yang telah disampaikan.

Tabel 5.12 Struktur Tabel siap_kategori

Nama Atribut	Tipe	Lebar	Keterangan
id_kategori	int	11	sebagai <i>primary key</i> dan <i>auto increment</i>
kategori	varchar	50	menyimpan nama kategori

Pada Tabel 5.12 menunjukkan struktur tabel siap_kategori. Tabel tersebut merupakan tabel yang menampung data jenis-jenis kategori aspirasi, ini digunakan untuk memisahkan/mengkategorikan aspirasi yang disampaikan sesuai ranah lingkungannya.

Tabel 5.13 Struktur Tabel siap_post

Nama Atribut	Tipe	Lebar	Keterangan
id_post	int	11	sebagai <i>primary key</i> dan <i>auto increment</i>
judul	varchar	30	menyimpan judul postingan sistem
konten	longtext		menyimpan isi/konten postingan
waktu	timestamp		bawaan = <i>current_timestamp</i>

Pada Tabel 5.13 di atas menunjukkan struktur tabel siap_post. Tabel tersebut merupakan tabel yang menampung data postingan yang terdapat pada halaman sistem, postingan tersebut meliputi halaman tentang sistem, petunjuk dan syarat, kontak dan konten-konten dari *slider* yang ditampilkan pada halaman awal sistem.

5.3.5 Perancangan dan Integrasi Komponen

Perancangan dan Integrasi komponen merupakan rancangan dari setiap komponen ulang yang diintegrasikan untuk membuat sistem baru. Menurut Sommerville (2011) perancangan berisi pernyataan sederhana bagaimana dari setiap fungsi program dapat dijalankan dan dapat diimplementasikan pada tahapan berikutnya. Integrasi komponen digunakan untuk mempermudah *programmer* dalam proses implementasi program karena tidak harus membuatnya dari awal. Perancangan dengan menggunakan komponen memuat algoritma yang akan diimplementasikan pada kelas-kelas yang didefinisikan. Beberapa perancangan dan integrasi komponen Sistem Informasi Aspirasi *Online* pada DPM UB berikut ini merupakan sampel dari fungsi dan komponen yang digunakan ulang pada kelas Aspirasi, User, Mahasiswa, Lembaga dan Admin.

5.3.5.1 Perancangan Komponen Fungsi Login Mahasiswa

Pada kelas Mahasiswa terdapat fungsi login() yang berfungsi untuk mengotentikasi akun mahasiswa. Pada fungsi ini menggunakan komponen *web service* (API) yang disediakan oleh *Brawijaya Authentication and Identification System* (BAIS). Tabel 5.14 berikut ini merupakan algoritme dari fungsi login yang terdapat dalam kelas *controller* Mahasiswa.

Tabel 5.14 Perancangan Komponen Fungsi Login Mahasiswa

```

fungsi login()
    variabel nim yang menerima masukan dari kolom nim di form login
    variabel pass yang menerima masukan dari kolom password di form login
    variabel login_check yang berisi kembalian dari hasil menjalankan
    fungsi BAIS([nim, pass]) yang terdapat pada kelas model REST_API
    untuk melakukan otentikasi pada bais.ub.ac.id/api.
    jika kembalian dari login_check[sukses] bernilai 'yes' maka:
    seluruh data yang terdapat pada variabel login_check dimasukkan
    kedalam session userdata
    mencetaknya json yang berisi kata 'yes'
    sedangkan jika login_check[sukses] bernilai selain kata 'yes' maka:
    variabel pesan1 berisi kalimat "Hai l[nama], Maaf (pesan
    kesalahan)!"
    variabel pesan2 berisi kalimat "Hai [nim], Maaf (pesan kesalahan)!"
  
```


Tabel 5.14 Perancangan Komponen Fungsi Login Mahasiswa (lanjutan)

```

variabel pesan sama dengan pesan2 jika [nama] bernilai false, namun
jika selain itu akan sama dengan pesan1
mencetak json yang berisi kalimat dari variabel pesan

```

Sedangkan pada kelas REST_API terdapat fungsi BAIS() yang akan menjadi penghubung dan juga sebagai penerima kembalian dari *web service* yang disediakan oleh *bais.ub.ac.id/api*. Berikut algoritmenya terdapat pada Tabel 5.15.

Tabel 5.15 Perancangan Komponen Fungsi Otentikasi BAIS

```

fungsi BAIS(data)
kondisi jika data tersedia nim dan password maka:
variabel ip diisi dengan ip server
variabel nim diisi dengan data nim
variabel password diisi dengan data password
lalu ketiga variabel tersebut dikirimkan ke bais.ub.ac.id/api
menggunakan library curl yang disediakan oleh PHP untuk mengambil
data melalui URL.
variabel result menyimpan kembalian dari hasil eksekusi menggunakan
curl.
menerjemahkan variabel result yang berisi string JSON menjadi
variabel PHP menggunakan json_decode.
selanjutnya dilakukan modifikasi untuk mengambil data-data yang
diperlukan saja, dan dimasukkan kedalam variabel array
melakukan return (pengembalian) data dari variabel array
kondisi jika data tidak tersedia nim dan password maka:
variabel data array diisi dengan string "ERROR"
melakukan return (pengembalian) data dari variabel array

```

5.3.5.2 Perancangan Komponen Fungsi Login Lembaga

Pada kelas Lembaga juga terdapat fungsi login() yang berfungsi untuk mengotentikasi akun lembaga yang sudah terdaftar didalam basis data. Berikut Tabel 5.16 merupakan algoritme dari fungsi login yang terdapat dalam kelas Lembaga. Algoritme ini hampir sama dengan aligoritme yang digunakan operator dan administrator untuk melakukan login.

Tabel 5.16 Perancangan Komponen Fungsi Login Lembaga

```

fungsi login()
variabel username yang menerima masukan dari kolom username di form
login
variabel password yang menerima masukan dari kolom password di form
login
variabel cekAkun yang berisi kembalian dari menjalankan fungsi
cekAkun(username) yang terdapat dalam model M_User
variabel login_check yang berisi kembalian dari hasil menjalankan
fungsi getLembaga(username, password) yang terdapat pada model
M_User untuk melakukan otentikasi dengan data yang terdapat pada
basis data.
jika kembalian dari login_check bernilai 'FALSE' maka:
jika cekAkun bernilai TRUE maka:
variabel pesan berisi kalimat "Hai [nama], Maaf Password anda
salah!"
sebaliknya jika cekAkun bernilai FALSE maka:
variabel pesan berisi kalimat "Hai [username], Maaf Username tidak
terdapat dalam database!"
mencetaknya json yang berisi kata kalimat dalam variabel pesan
namun jika login_check bernilai TRUE maka:

```

Tabel 5.16 Perancangan Komponen Fungsi Login Lembaga (lanjutan)

```

seluruh data yang terdapat pada variabel login_check dimasukkan
kedalam session userdata
mencetaknya json yang berisi kata 'yes'

```

5.3.5.3 Perancangan Komponen Fungsi Mengirimkan Aspirasi

Pada kelas Aspirasi terdapat fungsi kirim() yang berfungsi untuk memproses data aspirasi yang selanjutnya disimpan kedalam basis data sistem. Komponen yang digunakan ulang pada fungsi ini adalah *library* dari Ci PHP Qrcode yang digunakan untuk membuat gambar qrcode dan PHPMailer yang digunakan untuk mengirimkan email notifikasi kepada pengguna. Tabel 5.17 berikut ini merupakan algoritme dari fungsi kirim() yang terdapat dalam kelas Aspirasi.

Tabel 5.17 Perancangan Komponen Fungsi Mengirimkan Aspirasi

```

fungsi kirim()
melakukan pengecekan session[user_siap] jika bernilai TRUE maka:
jika tidak menerima kiriman dari method post[aspirasi] maka:
variabel id_aspirasi diisi dengan hasil kembalian dari fungsi
generateID
membuat path folder sesuai tahun dan bulan saai ini
melakukan pengecekan ketersediaan lampiran
mengunggah lampiran kedalam hosting
memanggil library ciqrcode
membuat gambar qrcode dan menyimpan ke dalam folder path
variabel data diisi dengan seluruh data aspirasi
memanggil fungsi addAspirasi(data) yang terdapat dalam model
M_Aspirasi
jika sukses memanggil fungsi addNotifikasi([id_aspirasi, username])
memanggil fungsi kirim_notifikasi() untuk mengirimkan email ke
pengguna
memindahkan halaman ke halaman detail aspirasi
namun jika tidak menerima kiriman dari method post[aspirasi] maka:
menampilkan view form aspirasi
jika session bernilai FALSE maka dialihkan ke halaman awal

```

5.3.5.4 Perancangan Komponen Fungsi Memantau Aspirasi

Pada kelas Aspirasi terdapat fungsi detail() yang berfungsi untuk menampilkan detail data aspirasi dari lembaga/mahasiswa dengan menggunakan masukan id aspirasi yang telah diinputkan atau dikirimkan pada halaman sebelumnya, sehingga dapat mencarinya dalam basis data menggunakan id aspirasi tersebut. Komponen Time Ago Helpers dan Mobile Detect digunakan pada fungsi ini namun diletakkan pada algoritme *view* program, bukan pada algoritme fungsi di *controller*. Tabel 5.18 berikut ini merupakan algoritme dari fungsi memantau aspirasi yang terdapat pada kelas *controller* sistem.

Tabel 5.18 Perancangan Komponen Fungsi Memantau Aspirasi

```

fungsi detail()
melakukan pengecekan ketersediaan data dari segment ketiga atau
kiriman dari method post yang berisi ID aspirasi, jika tersedia maka
mengisi variabel id_aspirasi dengan data yang didapatkan
jika tidak akan dialihkan ke halaman awal aspirasi
variabel data diisi dengan kembalian dari fungsi
getAspirasi(id_aspirasi) dalam model M_Aspirasi
jika terdapat data maka akan mengambil data-data lainnya
variabel kategori diisi dengan kembalian dari getKategori

```

Tabel 5.18 Perancangan Komponen Fungsi Memantau Aspirasi (lanjutan)

```

variabel status diisi dengan kembalian dari getStatus
variabel penerima diisi dengan kembalian dari getPenerima
variabel pengirim diisi dengan kembalian dari getPengirim
variabel tanggapan diisi dengan kembalian dari getTanggapan
lalu memanggil view Aspirasi_Detail

```

5.3.5.5 Perancangan Komponen Fungsi Memberikan Tanggapan

Pada kelas Aspirasi terdapat fungsi tambah_tanggapan() yang berfungsi untuk menyimpan tanggapan atas aspirasi yang disampaikan mahasiswa/lembaga. Komponen Time Ago Helpers juga terdapat pada *view* fungsi ini yang digunakan untuk mengkonversi waktu kedalam format yang disediakan Time Ago Helpers. Berikut ini merupakan algoritme dari fungsi memberikan tanggapan yang diuraikan pada Tabel 5.19.

Tabel 5.19 Perancangan Komponen Fungsi Memberikan tanggapan

```

fungsi tambah_tanggapan()
melakukan pengecekan session[user_siap] jika bernilai TRUE maka:
jika menerima kiriman dari method post[isi] maka
menyimpan data yang dikirimkan menggunakan fungsi addTanggapan yang
terdapat dalam model M_Aspirasi
memanggil view Aspirasi_Detail_Tanggapan
mengambil data aspirasi melalui fungsi getAspirasi(id aspirasi)
menambahkan notifikasi melaui fungsi addNotifikasi
jika yang memberikan tanggapan adalah akun lain maka
mengirimkan email notifikasi melalui fungsi kirim_notifikasi(data)
jika session bernilai FALSE maka dialihkan ke halaman awal

```

5.3.5.6 Perancangan Komponen Fungsi Mengelola Akun

Pada kelas Admin terdapat fungsi kelolaAkun() yang berfungsi untuk mengelola Akun yang terdaftar dalam basis data, meliputi: akun mahasiswa, lembaga, operator dan administrator. Pada fungsi kelolaAkun() aktor administrator dapat melakukan penambahan akun baru, edit dan hapus akun untuk operator dan lembaga, sedangkan untuk mahasiswa hanya dapat melihat daftarnya saja. Komponen yang digunakan dalam fungsi ini adalah logika dari *source code* LAPOR BEM dalam melakukan CRUD data. Tabel 5.20 berikut ini merupakan algoritme dari fungsi mengelola akun.

Tabel 5.20 Perancangan Komponen Fungsi Mengelola Akun

```

fungsi kelolaAkun()
melakukan pengecekan session[user_siap] dan session[full_akses] jika
bernilai TRUE maka:
mengambil data dari segment ketiga
jika segment ketiga bernilai edit maka
jika menerima kiriman dari method post[tipe] dan post[username]
mengambil data dari segment keempat dan kelima
jika data dari segment keempat bernilai admin maka
menyimpan data menggunakan fungsi updateAdmin yang terdapat dalam
model M_User
jika data dari segment keempat bernilai lembaga maka
menyimpan data menggunakan fungsi updateLembaga yang terdapat dalam
model M_User
lalu dikembalikan kehalaman kelolaAkun
jika tidak menerima kiriman dari method post maka

```

Tabel 5.20 Perancangan Komponen Fungsi Mengelola Akun (lanjutan)

```

mengambil data menggunakan fungsi cekAkun(username) dalam model
M_User
lalu menampilkan view Admin_KelolaAkun_Edit
namun jika segment ketiga bernilai hapus maka
mengambil data dari segment keempat dan kelima
melakukan penghapusan data menggunakan fungsi deleteAkun(data)
namun jika segment ketiga bernilai selain sebelumnya maka
mengambil seluruh data pengguna menggunakan fungsi getUsers(tipe)
dari model M_User
lalu memanggil view Admin_KelolaAkun
jika kondisi session tidak terpenuhi dialihkan ke halaman awal

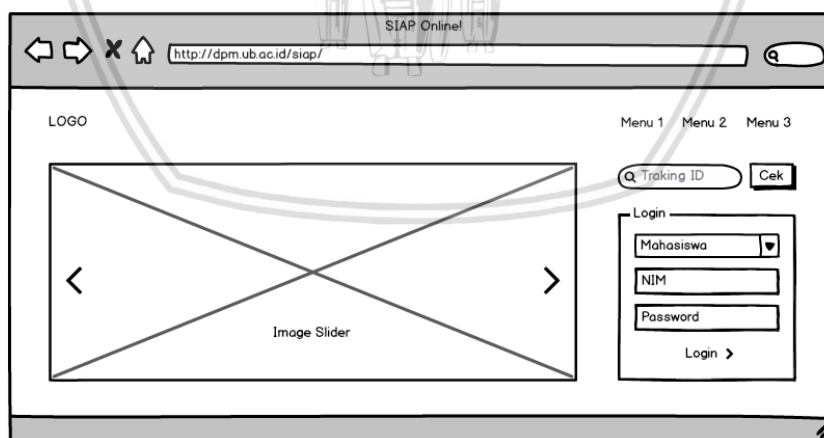
```

CodeIgniter yang merupakan *back-end framework* digunakan untuk keseluruhan implementasi fungsi sistem, sedangkan untuk *front-end framework* yang digunakan adalah AdminLTE-3.0 yang akan dijelaskan pada tahap perancangan antarmuka pengguna.

5.3.6 Perancangan Antarmuka

Perancangan antarmuka pengguna dilakukan untuk menggambarkan tampilan dari sistem yang akan dilihat oleh pengguna secara langsung, perancangan antarmuka ini berisi beberapa sketsa tampilan antarmuka pengguna Sistem Informasi Aspirasi *Online* pada DPM UB yang dikembangkan dalam penelitian ini. Beberapa sketsa antarmuka sistem sebagai sampel diantaranya adalah antarmuka halaman utama, profil, kirim aspirasi, statistik aspirasi, kelola aspirasi, detail aspirasi dan kelola akun. Sketsa-sketsa berikut ini merupakan hasil dari penyesuaian dengan *layout* yang disediakan oleh *front-end framework* AdminLTE-3.0.

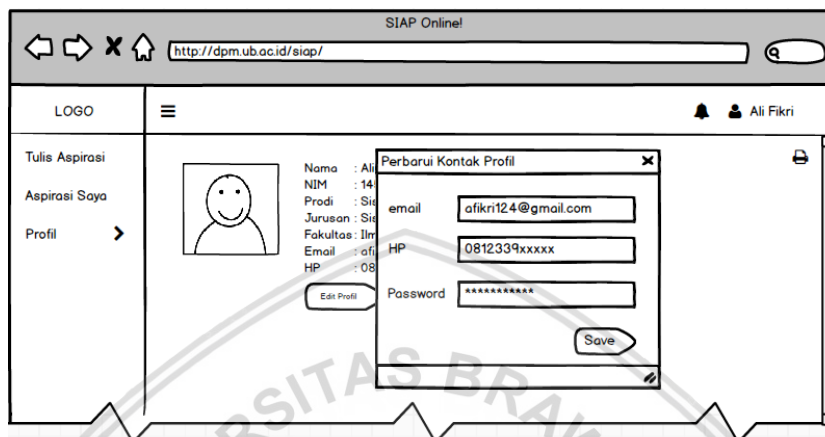
5.3.6.1 Rancangan Halaman Utama

**Gambar 5.11 Rancangan Halaman Utama**

Pada Gambar 5.11 merupakan gambaran halaman utama sistem, halaman ini adalah halaman pertama yang akan dikunjungi oleh visitor ataupun pengguna. Pada halaman ini terdapat dua fungsi utama yang dapat dijalankan, yaitu fungsi untuk tracking ID aspirasi atau memantau perkembangan aspirasi menggunakan ID aspirasi yang telah didapatkan sebelumnya. Fungsi yang kedua adalah tampilan

form untuk masuk (*login*) ke dalam sistem. Tampilan *login* ini akan berubah jika status aktor dalam keadaan masuk (*logged in*) sistem, tampilan yang berubah terdapat pada *form login* yang diganti dengan tampilan singkat nama dan nim yang sedang dalam keadaan *login* dan untuk tombol *Login* diganti dengan tombol Tulis Aspirasi.

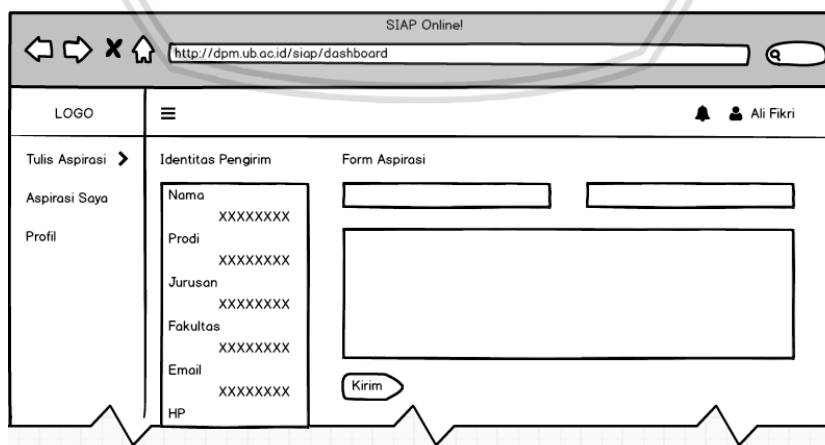
5.3.6.2 Rancangan Halaman Profil



Gambar 5.12 Rancangan Halaman Profil

Gambar 5.12 merupakan gambaran dari halaman profil yang dapat digunakan untuk memperbarui identitas akun, halaman ini akan diakses otomatis jika pengguna baru pertama kali *login* sistem. Karena terdapat data yang tidak didapatkan dari API BAIS ataupun dari data yang didaftarkan oleh administrator namun dibutuhkan sistem sehingga data belum lengkap. Aktor harus melengkapi data untuk dapat melanjutkan aktivitasnya. Halaman ini juga dapat diakses sewaktu-waktu jika pengguna ingin memperbarui datanya dengan cara mengakses menu profil yang terdapat pada *sidebar* sistem.

5.3.6.3 Rancangan Halaman Kirim Aspirasi

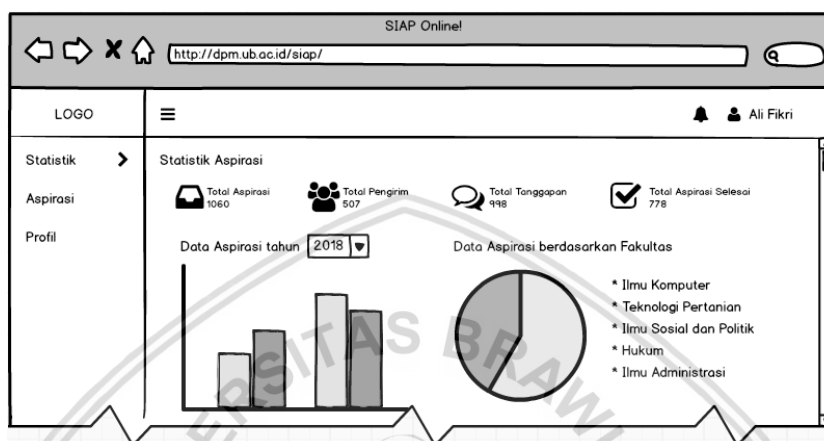


Gambar 5.13 Rancangan Halaman Kirim Aspirasi

Gambar 5.13 merupakan gambaran dari halaman Kirim Aspirasi, dimana halaman ini merupakan tujuan utama dari aktor Mahasiswa/Lembaga untuk

menyampaikan aspirasinya secara *online* tanpa perlu mendatangi sekretariat DPM UB. Pada Halaman ini terdapat identitas dari aktor yang sedang *login* selaku pengirim dan *form* Aspirasi yang terdiri dari subjek, kategori, konten/isi dan lampiran. Data tersebut akan disimpan ke dalam basis data untuk dikelola oleh operator/administrator untuk didisposisikan atau disampaikan ke pihak-pihak yang terkait dengan permasalahannya.

5.3.6.4 Rancangan Halaman Statistik Aspirasi

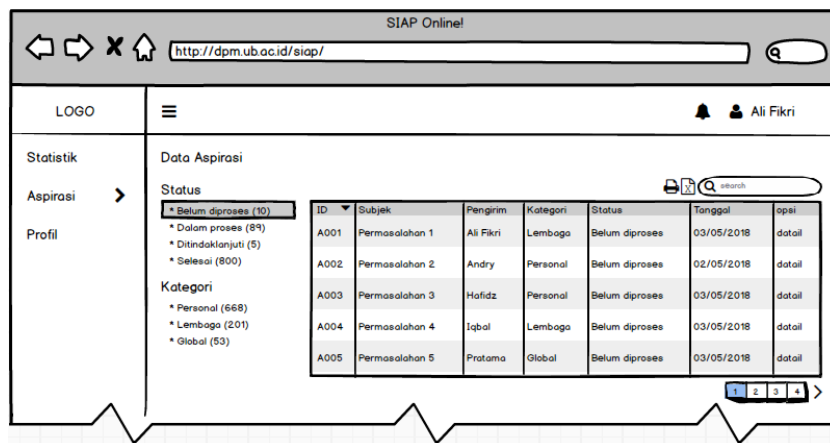


Gambar 5.14 Rancangan Halaman Statistik Aspirasi

Gambar 5.14 merupakan gambaran dari halaman Statistik Aspirasi, dimana halaman ini merupakan ringkasan/rekapan data aspirasi yang telah diterima sistem, halaman ini dapat diakses oleh operator dan administrator selaku penanggung jawab dan pengelola aspirasi. Didalam halaman ini data aspirasi disajikan kedalam bentuk *chart* sehingga aktor dapat mengetahui dengan mudah hasil statistik dari aspirasi. Statistik dalam halaman ini menampilkan jumlah data aspirasi yang masuk perbulan dalam periode tahun tertentu, data jumlah aspirasi juga dikelompokkan berdasarkan fakultas yang dikirimkan oleh mahasiswa, tidak untuk aspirasi yang dikirimkan oleh lembaga. Total Aspirasi, pengirim, tanggapan dan aspirasi yang sudah terselesaikan juga ditampilkan dalam halaman ini.

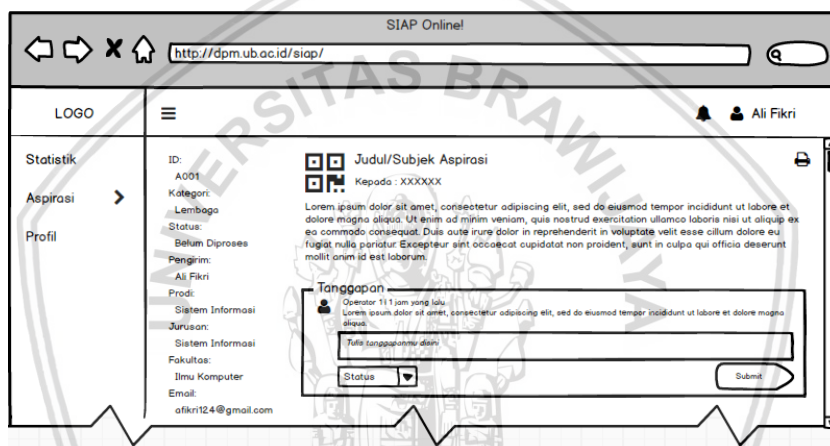
5.3.6.5 Rancangan Halaman Kelola Data Aspirasi

Gambar 5.15 merupakan gambaran dari halaman kelola data aspirasi, dimana halaman ini menampilkan data aspirasi ke dalam bentuk tabel yang dapat di *filter* berdasarkan status atau kategori. Data pada tabel akan disajikan kedalam beberapa halaman (*page*), untuk melakukan tindakan pada setiap aspirasi aktor dapat menekan fungsi detail. Pada halaman ini, aktor juga dapat melakukan pencarian data, cetak dan ekspor ke dalam format *excel*. Pengklasifikasian data dapat dilakukan melalui halaman ini dengan mengaktifkan klasifikasi berdasarkan filter yang diaktifkan.



Gambar 5.15 Rancangan Halaman Kelola Data Aspirasi

5.3.6.6 Rancangan Halaman Detail Aspirasi

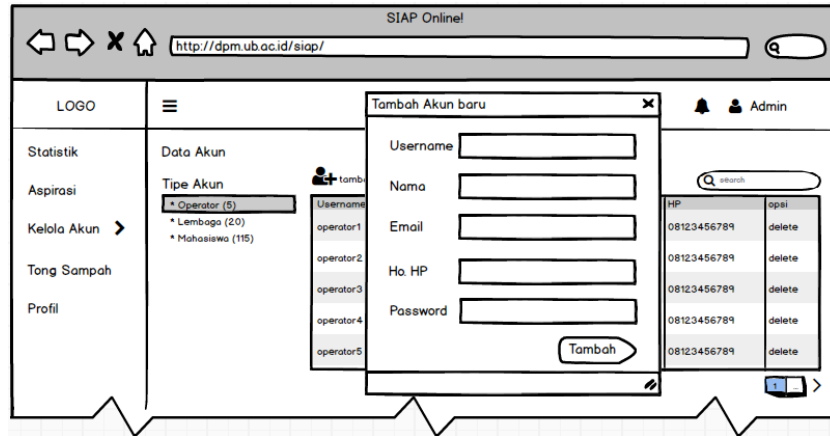


Gambar 5.16 Rancangan Halaman Detail Aspirasi

Pada Gambar 5.16 merupakan gambaran dari halaman detail aspirasi yang dapat digunakan untuk memantau aspirasi yang sudah dikirimkan, halaman ini menampilkan informasi pengirim dan isi/konten aspirasi secara lebih detail beserta statusnya. Aktor dapat memberikan tanggapan pada setiap aspirasi dan untuk operator dapat mengganti statusnya melalui halaman ini, halaman ini juga menyediakan fungsi cetak untuk melakukan pencetakan terhadap detail aspirasi yang sedang ditampilkan sebagai bukti yang sah.

5.3.6.7 Rancangan Halaman Kelola Akun

Halaman ini menampilkan data akun yang terdaftar pada sistem dan halaman ini hanya dapat diakses oleh aktor administrator selaku pengguna sistem yang mempunyai hak akses tertinggi. Data akun ditampilkan kedalam bentuk tabel yang dikategorikan berdasarkan tipe akun, fungsi yang dapat dilakukan aktor pada halaman ini adalah untuk menambah akun baru, edit akun dan menghapus akun yang sudah ada. Fungsi tambah akun baru yang berisi *form* pendaftaran ditampilkan pada halaman yang sama, namun ditampilkan ke dalam bentuk *pop-up* seperti yang digambarkan pada Gambar 5.17 berikut ini.



Gambar 5.17 Rancangan Halaman Kelola Akun



BAB 6 IMPLEMENTASI

Implementasi berisi penjelasan mengenai hasil kebutuhan implementasi Sistem Informasi Aspirasi *Online* pada DPM UB yang dikembangkan dalam penelitian ini. Hal-hal yang dijelaskan dalam tahapan ini diantaranya adalah spesifikasi lingkungan implementasi, Implementasi beberapa algoritma kode program dan beberapa gambar antarmuka pengguna. Hasil dari analisis persyaratan, analisis komponen dan perancangan pada tahapan sebelumnya menjadi acuan dalam melakukan implementasi sistem yang dikembangkan. Hasil implementasi pada tahap ini diharapkan mampu menjawab kebutuhan pengguna yang telah dipaparkan pada bab sebelumnya.

6.1 Spesifikasi Lingkungan Implementasi

Spesifikasi lingkungan implementasi sistem menjelaskan bagaimana spesifikasi sebuah perangkat keras dan perangkat lunak yang digunakan selama proses pengembangan dilakukan. Perangkat keras yang digunakan pada proses pembangunan sistem informasi ini adalah sebuah unit laptop dengan spesifikasi yang dijelaskan pada Tabel 6.1. Sedangkan untuk spesifikasi perangkat lunak yang digunakan dalam pengembangan ditunjukkan pada Tabel 6.2.

Tabel 6.1 Spesifikasi Perangkat Keras

Unit Komputasi	Laptop ASUS A455L
CPU	Intel(R) Core(TM) i5-4210U 2.39 Ghz
Kapasitas RAM	4096 MB
Kapasitas Penyimpanan	1 TB
Kartu Grafis	NVidia GeForce 840M
Resolusi Layar	1366 x 768 pixels

Tabel 6.2 Spesifikasi Perangkat Lunak

Sistem Operasi	Windows 10 Enterprise 64-bit
Web Server	Xampp v3.2.2
Web Browser	Google Chrome Version 67.0.3396.99 (64-bit)
Editor Kode Program	Sublime Text v3, Build 3143
UML Designer	Astah Model version 37
Dokumentasi	MS Word 2016

6.2 Implementasi dan Integrasi Algoritma

Pada implementasi dan integrasi algoritma program dalam penelitian ini menerapkan pola perancangan *model-view-controller (MVC)*. Karena menerapkan metode *reuse-oriented development* maka dalam proses pengembangannya menggunakan algoritma program yang sudah ada untuk digunakan ulang. Komponen utama yang digunakan dalam pengembangan adalah *framework* Codeigniter yang merupakan kerangka algoritma *back-end* pemrograman berbasis web. Dari komponen-komponen yang telah dianalisis dilakukan integrasi untuk menjadi satu kesatuan dalam membentuk Sistem Informasi Aspirasi *Online* pada DPM UB. Fungsi yang diuraikan dalam bab ini hanya beberapa fungsi saja dari keseluruhan fungsi sistem, fungsi yang akan diuraikan merupakan fungsi yang terdapat pada kelas *controller* dan *model* sistem. Komponen yang digunakan ulang dan modifikasi yang dilakukan pada komponen juga dijelaskan pada tahapan ini.

6.2.1 Implementasi Fungsi Login Mahasiswa

Untuk mengidentifikasi keaslian akun mahasiswa dalam melakukan login pada sistem, maka diperlukan otentikasi menggunakan akun yang sama dengan yang mahasiswa gunakan untuk login pada sistem-sistem lain semisal Sistem Informasi Akademik Mahasiswa Universitas Brawijaya (SIAM UB). Akun tersebut merupakan akun yang disediakan oleh *Brawijaya Authentication and Identification System (BAIS) UB*. Oleh karena itu, pada tahap implementasi sistem ini juga menggunakan komponen dari layanan *web service* yang disediakan oleh BAIS UB.

Tabel 6.3 berikut ini merupakan hasil implementasi dan integrasi kode program yang terdapat pada kelas *controller* Mahasiswa. Algoritma kode program ini digunakan untuk pengecekan data, login dan menyimpan data mahasiswa kedalam basis data.

Tabel 6.3 Implementasi Kelas Controller Mahasiswa

No	Kode Program
1	<code><?php</code>
2	<code>defined('BASEPATH') OR exit('No direct script access allowed');</code>
3	<code>require_once APPPATH . 'controllers/User.php';</code>
4	<code>class Mahasiswa extends User {</code>
5	<code>protected \$prodi;</code>
6	<code>protected \$jurusan;</code>
7	<code>protected \$fakultas;</code>
8	<code>public function __construct(){</code>
9	<code>parent::__construct();</code>
10	<code>\$this->load->model('REST_API');</code>
11	<code>}</code>
12	<code>public function index(){</code>
13	<code>if (\$this->session->userdata('nim')) {</code>
14	<code>\$username = \$this->session->userdata('nim');</code>
15	<code>\$cekAkun = \$this->M_User->cekAkun(\$username);</code>
16	<code>if (sizeof(\$cekAkun) == 0) {</code>
17	<code>\$this->tambahAkun();</code>
18	<code>} elseif(\$cekAkun[0]['verifikasi'] == 0){</code>
19	<code>\$data = \$cekAkun[0];</code>
20	<code>\$this->session->set_userdata(\$data);</code>
21	<code>\$this->konfirmasiEmail();</code>
22	<code>} else {</code>

Tabel 6.3 Implementasi Kelas Controller Mahasiswa

No	Kode Program
23	<code>\$data = array('user_siap' => true,</code>
24	<code> 'img' =></code>
25	<code>"http://siakad.ub.ac.id/siam/██████████" *</code>
26	<code> \$getAkun = \$this->M_User->getAkunMahasiswa(\$username);</code>
27	<code> \$session_user = array_merge(\$data, \$getAkun);</code>
28	<code> \$this->session->set_userdata(\$session_user);</code>
29	<code> redirect(base_url(), 'refresh');</code>
30	<code> }</code>
31	<code> } else {</code>
32	<code> redirect(base_url(), 'refresh');</code>
33	<code> }</code>
34	<code> }</code>
35	<code> public function login(){</code>
36	<code> \$data['nim'] = \$this->input->post('uid');</code>
37	<code> \$data['pass'] = \$this->input->post('passwd');</code>
38	<code> \$login_check = \$this->REST_API->BAIS(\$data);</code>
39	<code> if (\$login_check['sukses'] == 'yes') {</code>
40	<code> \$this->session->set_userdata(\$login_check);</code>
41	<code> echo json_encode('yes');</code>
42	<code> } else {</code>
43	<code> \$pesan1 = "Hai</code>
44	<code> ".ucwords(strtolower(\$login_check['nama'])).", Maaf</code>
45	<code> ".\$login_check['sukses']. "!";</code>
46	<code> \$pesan2 = "Hai ".ucwords(strtolower(\$login_check['nim'])).",</code>
47	<code> Maaf ".\$login_check['sukses']. "!";</code>
48	<code> \$pesan = (\$login_check['nama'] == false ? \$pesan2:</code>
49	<code> \$pesan1);</code>
50	<code> echo json_encode(\$pesan);</code>
51	<code> }</code>
52	<code> }</code>
53	<code> public function tambahAkun() {</code>
54	<code> \$data = array('nim' => \$this->session->userdata('nim'),</code>
55	<code> 'nama' => \$this->session->userdata('nama'),</code>
56	<code> 'prodi' => \$this->session->userdata('prodi'),</code>
57	<code> 'jurusan' => \$this->session->userdata('jurusan'),</code>
58	<code> 'fakultas' => \$this->session->userdata('fakultas'),</code>
59	<code> 'tipe' => 'mahasiswa');</code>
60	<code> if (\$this->input->post('email') != NULL) {</code>
61	<code> \$user = array('username' => \$data['nim'],</code>
62	<code> 'nama' => \$data['nama'],</code>
63	<code> 'email' => \$this->input->post('email'),</code>
64	<code> 'hp' => \$this->input->post('hp'),</code>
65	<code> 'tipe' => "mahasiswa");</code>
66	<code> \$mahasiswa = array('username' => \$data['nim'],</code>
67	<code> 'prodi' => \$data['prodi'],</code>
68	<code> 'jurusan' => \$data['jurusan'],</code>
69	<code> 'fakultas' => \$data['fakultas']);</code>
70	<code> \$insertToDB = \$this->M_User->addMahasiswa(\$user,</code>
71	<code> \$mahasiswa);</code>
72	<code> \$dataEmail = array('email' => \$user['email'], 'nama' =></code>
73	<code> \$user['nama'], 'username' => \$user['username']);</code>
74	<code> \$this-> kirimEmailKonfirmasi(\$dataEmail);</code>
75	<code> redirect(base_url(). 'mahasiswa', 'refresh');</code>
76	<code> } else {</code>
77	<code> \$this->load->view('Mahasiswa_TambahAkun', \$data);</code>
78	<code> }</code>
79	<code> }</code>
80	<code>}</code>

Catatan: * Kode program bersifat rahasia.

Dalam Tabel 6.3, pada baris ke-10 terdapat kode program `$this->load->model('REST_API')` kode ini merupakan pemanggilan kelas REST_API yang terdapat dalam model, kemudian pada baris ke-38 terdapat kode `$this->REST_API->BAIS($data)` kode ini merupakan pemanggilan fungsi BAIS yang terdapat dalam kelas model REST_API. Hasil implementasi dan integrasi algoritma pada fungsi BAIS diuraikan dalam Tabel 6.4, fungsi algoritma dalam tabel tersebut digunakan untuk melakukan otentikasi akun mahasiswa menggunakan komponen *web service* yang disediakan BAIS UB, kembalian dari API BAIS dalam bentuk String JSON yang terdiri dari kumpulan objek-objek yang berisi data mahasiswa, sehingga diperlukan modifikasi untuk mengambil data yang dibutuhkan saja. Selanjutnya, data tersebut dikembalikan ke *controller* dalam bentuk array.

Tabel 6.4 Implementasi Kelas Model REST_API

No	Kode Program
1	<?php
2	defined('BASEPATH') OR exit('No direct script access allowed');
3	class REST_API extends CI_Model {
4	public function BAIS(\$data) {
5	if(isset(\$data['nim']) && isset(\$data['pass'])) {
6	\$ip = \$_SERVER['REMOTE_ADDR'];
7	\$nim = \$data['nim'];
8	\$password = [REDACTED] *
9	\$link =
10	'https://bais.ub.ac.id/api/[REDACTED]'
11	\$ch = curl_init();
12	curl_setopt(\$ch, CURLOPT_SSL_VERIFYPEER, false);
13	curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true);
14	curl_setopt(\$ch, CURLOPT_URL, \$link);
15	\$result = curl_exec(\$ch);
16	curl_close(\$ch);
17	\$obj = json_decode(\$result);
18	\$arr = (array) \$obj;
19	\$sukses = (\$arr['error'] == 0 ? 'yes' : \$arr['errmsg']);
20	\$nama = ucwords(strtolower(\$arr['USER']->NAMA));
21	\$data = array('sukses' => \$sukses, 'nim' => \$arr['USER']->
22	ID, 'nama' => \$nama, 'fakultas' => \$arr['CONTENT']->FAKULTAS,
23	'jurusan' => \$arr['CONTENT']->JURUSAN, 'prodi' => \$arr['CONTENT']->
24	PRODI);
25	return \$data;
26	} else {
27	\$data = array('sukses' => "ERROR API");
28	return \$data;
29	}
30	}
31	}

Catatan: * Kode program bersifat rahasia.

6.2.2 Implementasi Fungsi Login Lembaga

Dalam melakukan indentifikasi akun lembaga, identifikasi yang dilakukan berbeda dengan identifikasi yang dilakukan pada aktor mahasiswa. Aktor lembaga tidak memerlukan otentikasi melalui *web service* BAIS, melainkan hanya mencocokkan data dengan yang ada di basis data sistem. Berikut ini pada Tabel 6.5 merupakan hasil implementasi dari fungsi login yang terdapat pada kelas

controller Mahasiswa, sedangkan untuk fungsi lain tidak disertakan pada tabel tersebut.

Tabel 6.5 Implementasi Kelas Controller Lembaga

No	Kode Program
1	<?php
2	defined('BASEPATH') OR exit('No direct script access allowed');
3	require_once APPPATH . 'controllers/User.php';
4	class Lembaga extends User {
5	protected \$password;
6	protected \$alamat;
7	protected \$logo;
8	public function __construct(){
9	parent::__construct();
10	}
11	public function login(){
12	\$this->username = htmlentities(\$this->input->post('uid'));
13	\$this->password = htmlentities(\$this->input->post('passwd'));
14	\$cekAkun = \$this->M_User->cekAkun(\$this->username);
15	\$login_check = \$this->M_User->getLembaga(\$this->username,
16	\$this->password);
17	if (!\$login_check) {
18	if (\$cekAkun) {
19	\$pesan = "Hai ". \$cekAkun[0]['nama'].", Maaf Password anda
20	salah!";
21	} else {
22	\$pesan = "Hai ". \$this->username.", Maaf Username tidak
23	terdapat dalam database!";
24	}
25	echo json_encode(\$pesan);
26	} else {
27	\$this->session->set_userdata(\$login_check);
28	echo json_encode('yes');
29	}
30	}
31	}

6.2.3 Implementasi Fungsi Mengirimkan Aspirasi

Dalam melakukan implementasi fungsi mengirimkan aspirasi komponen *reuse* (yang digunakan ulang) adalah *library* Ci PHP Qrcode dan PHPMailer. Ci PHP Qrcode yang digunakan untuk menghasilkan gambar qrcode yang dapat digunakan pengguna untuk alternatif mengakses detail aspirasi dengan melakukan *scan* menggunakan kamera ponsel pintar atau semacamnya. Sedangkan untuk PHPMailer digunakan untuk mengirimkan email notifikasi kepada pengguna sistem, dalam hal ini untuk mengirimkan ID aspirasi ke email pengirim aspirasi. Dalam Tabel 6.6 berikut ini menguraikan implementasi dan integrasi kode program fungsi mengirimkan aspirasi yang terdapat dalam *controller* Aspirasi.

Tabel 6.6 Implementasi Fungsi Mengirimkan Aspirasi

No	Kode Program
1	public function kirim(){
2	if (\$this->session->userdata('user_siap')) {
3	if (\$this->input->post('aspirasi') != NULL) {
4	\$lampiran = NULL;
5	\$qrcode = NULL;
6	\$id_aspirasi = \$this->generateID();

Tabel 6.6 Implementasi Fungsi Mengirimkan Aspirasi

No	Kode Program
7	<code>\$dt = new DateTime(null, new</code>
8	<code>DateTimeZone('Asia/Jakarta'));</code>
9	<code>\$thn = \$dt->format('Y');</code>
10	<code>\$bln = \$dt->format('m');</code>
11	<code>\$path = "assets/img/\$thn/\$bln/";</code>
12	<code>if (!file_exists(\$path)) {</code>
13	<code>if</code>
14	<code>(!file_exists("assets/img/\$thn/")){mkdir("assets/img/\$thn");}</code>
15	<code>mkdir("assets/img/\$thn/\$bln");</code>
16	<code>}</code>
17	<code>if (\$_FILES AND \$_FILES['lampiran']['name']) {</code>
18	<code>\$config['upload_path'] = ".\$path;</code>
19	<code>\$config['allowed_types'] = 'jpg png';</code>
20	<code>\$config['max_size'] = 3000;</code>
21	<code>\$this->load->library('upload', \$config);</code>
22	<code>if (! \$this->upload->do_upload('lampiran')){</code>
23	<code>\$pesan = \$this->upload->display_errors('','')." !";</code>
24	<code>echo "<script>alert('\$pesan');</script>";</code>
25	<code>redirect(base_url()).'aspirasi/kirim','refresh');</code>
26	<code>} else {</code>
27	<code>\$data = array('upload_data' => \$this->upload->data());</code>
28	<code>\$lampiran = \$path.\$data['upload_data']['file_name'];</code>
29	<code>}</code>
30	<code>}</code>
31	<code>\$this->load->library('ciqrcline');</code>
32	<code>\$config['cacheable'] = true;</code>
33	<code>\$config['cachedir'] = './assets/';</code>
34	<code>\$config['errorlog'] = './assets/';</code>
35	<code>\$config['imagedir'] = ".\$path;</code>
36	<code>\$config['quality'] = true;</code>
37	<code>\$config['size'] = '1024';</code>
38	<code>\$config['black'] = array(224,255,255);</code>
39	<code>\$config['white'] = array(70,130,180);</code>
40	<code>\$this->ciqrcline->initialize(\$config);</code>
41	<code>\$image_name = 'qr_'.\$id_aspirasi.'.png';</code>
42	<code>\$params['data'] =</code>
43	<code>base_url()).'aspirasi/detail/\$id_aspirasi';</code>
44	<code>\$params['level'] = 'H';</code>
45	<code>\$params['size'] = 10;</code>
46	<code>\$params['savename'] =</code>
47	<code>FCPATH.\$config['imagedir'].\$image_name;</code>
48	<code>if (\$this->ciqrcline->generate(\$params)){</code>
49	<code>\$qrcline = \$path.\$image_name;</code>
50	<code>}</code>
51	<code>\$data = array('id_aspirasi' => \$id_aspirasi,</code>
52	<code>'subjek' => htmlentities(\$this->input-</code>
53	<code>>post('subjek')),</code>
54	<code>'aspirasi' => htmlentities(\$this->input-</code>
55	<code>>post('aspirasi')),</code>
56	<code>'lampiran' => \$lampiran,</code>
57	<code>'qrcline' => \$qrcline,</code>
58	<code>'username' => \$this->session->userdata('username'),</code>
59	<code>'id_kategori' => \$this->input->post('kategori'));</code>
60	<code>if (\$this->M_Aspirasi->addAspirasi(\$data)){</code>
61	<code>\$this->M_Aspirasi->addNotifikasi(array(</code>
62	<code>'id_aspirasi'=> \$data['id_aspirasi'],</code>
63	<code>'username' => 'admin'));</code>
64	<code>\$data_pengirim = \$this->M_Aspirasi->getPengirim(array(</code>
65	<code>'username' => \$data['username']));</code>
66	<code>if (isset(\$data_pengirim['email'])) {</code>
67	

Tabel 6.6 Implementasi Fungsi Mengirimkan Aspirasi (lanjutan)

No	Kode Program
68	<code>\$data_email = array('email' =></code>
69	<code>\$data_pengirim['email'],</code>
70	<code>'nama' => \$data_pengirim['nama'],</code>
71	<code>'id_aspirasi' => \$id_aspirasi,</code>
72	<code>'status' => 'kirim');</code>
73	<code>\$this->kirim_notifikasi(\$data_email);</code>
74	<code>}</code>
75	
76	<code>redirect(base_url()."aspirasi/detail/\$id_aspirasi",'refresh');</code>
77	<code>} else {</code>
78	<code>echo "<script>alert('Terjadi masalah, coba</code>
79	<code>lagi!');</script>";</code>
80	<code>redirect(base_url().'aspirasi/kirim','refresh');</code>
81	<code>}</code>
82	<code>} else {</code>
83	<code>\$data['page'] = "Kirim";</code>
84	<code>\$data['pengirim'] = \$this->session->userdata();</code>
85	<code>\$data['kategori'] = \$this->M_Aspirasi-</code>
86	<code>>getKategori(array('id_kategori' => NULL));</code>
87	<code>\$this->load->view('Aspirasi_Kirim', \$data);</code>
88	<code>}</code>
89	<code>} else {</code>
90	<code>redirect(base_url(),'refresh');</code>
91	<code>}</code>
92	<code>}</code>

Dapat dilihat pada baris ke-31 sampai baris ke-50, kode program tersebut merupakan komponen yang digunakan ulang dan diintegrasikan untuk menghasilkan gambar QR-Code. Sedangkan untuk PHPMailer tidak terdapat secara langsung dalam fungsi ini, karena penggunaan *library* PHPMailer berada di dalam fungsi `$this->kirim_notifikasi($data_email)` yang dipanggil pada baris ke-65.

6.2.4 Implementasi Fungsi Memantau Aspirasi

Implementasi fungsi memantau aspirasi berfungsi untuk menampilkan detail data aspirasi berdasarkan ID aspirasi yang dikirimkan pada halaman sebelumnya. Terdapat dua cara untuk menampilkan detail aspirasi, yaitu dengan menggunakan *link* (meletakkan ID aspirasi pada *segment* ketiga) atau mengirimkannya melalui *method post*. ID yang diterima dicocokkan dengan data yang ada di basis data sistem, jika tersedia akan ditampilkan detail datanya, namun jika tidak akan menampilkan pesan kesalahan. Komponen yang digunakan ulang dalam fungsi ini adalah Time Ago Helpers dan Mobile Detect. Kedua komponen tersebut terdapat dalam *view* program, sehingga tidak terdapat pada Tabel 6.7 berikut ini yang merupakan implementasi dan integrasi kode program pada fungsi detail() dalam kelas *controller* Aspirasi.

Tabel 6.7 Implementasi Fungsi Memantau Aspirasi

No	Kode Program
1	<code>public function detail(){</code>
2	<code>if (\$this->uri->segment(3)) {</code>
3	<code>\$this->id_aspirasi = \$this->uri->segment(3);</code>

Tabel 6.7 Implementasi Fungsi Memantau Aspirasi (lanjutan)

No	Kode Program
4	} else if (\$this->id_aspirasi = \$this->input->post('id')){
5	\$this->id_aspirasi = \$this->input->post('id');
6	} else {
7	redirect(base_url().'aspirasi','refresh');
8	}
9	\$data['page'] = "Detail";
10	\$data['data'] = \$this->M_Aspirasi-
11	>getAspirasi(array('id_aspirasi' => \$this->id_aspirasi))[0];
12	if (\$data['data']['id_aspirasi'] != NULL) {
13	\$username = \$this->session->userdata('username');
14	if (\$this->uri->segment(3)){
15	if (\$data['data']['username'] == \$username){
16	\$this->M_Aspirasi->setNotifikasi(array('dibaca' => TRUE
17),
18	array('id_aspirasi' => \$this->uri->segment(3),
19	'username' => \$data['data']['username']));
20	} else if (\$this->session->userdata('tipe') == 'admin'){
21	\$this->M_Aspirasi->setNotifikasi(array('dibaca' => TRUE
22),
23	array('id_aspirasi' => \$this->uri->segment(3),
24	'username' => 'admin'));
25	}
26	}
27	\$data['kategori'] = \$this->M_Aspirasi-
28	>getKategori(array('id_kategori' =>
29	\$data['data']['id_kategori']))[0];
30	\$data['status'] = \$this->M_Aspirasi-
31	>getStatus(array('id_status' => \$data['data']['id_status']))[0];
32	\$data['status2'] = \$this->M_Aspirasi-
33	>getStatus(array('id_status' => Null));
34	\$data['penerima'] = \$this->M_Aspirasi-
35	>getPenerima(array('id_penerima' =>
36	\$data['data']['id_penerima']))[0];
37	\$data['penerima2'] = \$this->M_Aspirasi-
38	>getPenerima(array('id_penerima' => Null));
39	\$data['pengirim'] = \$this->M_Aspirasi-
40	>getPengirim(array('username' => \$data['data']['username']));
41	\$data['tanggapan'] = \$this->M_Aspirasi-
42	>getTanggapan(array('id_aspirasi' =>
43	\$data['data']['id_aspirasi']));
44	}
45	\$this->load->view('Aspirasi_Detail', \$data);
46	}

Fungsi yang disediakan *Time Ago Helpers* merupakan fungsi yang dapat digunakan ulang (*function reuse*) untuk menampilkan waktu kedalam format seperti di *facebook* dan *twitter*, contoh: “1 Jam yang lalu”, “2 Hari yang lalu” dan “baru saja”. Namun komponen yang tersedia hanya dalam format bahasa Inggris, sehingga perlu modifikasi pada komponen untuk menjadikan komponen tersebut berbahasa Indonesia. Sedangkan untuk komponen *Mobile Detect* digunakan untuk mendeteksi perangkat yang mengakses sistem, untuk memberikan tampilan yang sesuai dengan perangkat tersebut. Tabel 6.8 berikut ini merupakan hasil modifikasi komponen *Time Ago Helpers* yang digunakan pada sistem yang sedang dikembangkan, sedangkan untuk komponen yang belum dimodifikasi dapat dilihat di sumbernya yaitu pada *bismalabs.co.id*. Komponen ini tidak hanya digunakan pada fungsi detail aspirasi tapi digunakan juga pada beberapa fungsi yang lain.

Tabel 6.8 Implementasi Fungsi Time Ago Helpers (lanjutan)

No	Kode Program
1	<?php
2	if(!function_exists('time_to_ago')) {
3	function time_to_ago(\$datetime, \$full = false) {
4	\$now = new DateTime(null, new DateTimeZone('Asia/Jakarta'));
5	\$now = \$now->format('Y-m-d h:i:s');
6	\$dt = new DateTime(\$datetime);
7	\$dt = \$dt->format('Y-m-d h:i:s');
8	\$datediff = abs(strtotime(\$dt)-strtotime(\$now));
9	\$difftext="";
10	\$years = floor(\$datediff / (365*60*60*24));
11	\$months = floor((\$datediff - \$years * 365*60*60*24) /
12	(30*60*60*24));
13	\$days = floor((\$datediff - \$years * 365*60*60*24 -
14	\$months*30*60*60*24) / (60*60*24));
15	\$hours= floor(\$datediff/3600);
16	\$minutes= floor(\$datediff/60);
17	if(\$difftext==""){
18	if(\$years>=1)
19	\$difftext=\$years." tahun yang lalu";
20	}
21	if(\$difftext==""){
22	if(\$months>=1)
23	\$difftext=\$months." bulan yang lalu";
24	}
25	if(\$difftext==""){
26	if(\$days>=1)
27	\$difftext=\$days." hari yang lalu";
28	}
29	if(\$difftext==""){
30	if(\$hours>=1)
31	\$difftext=\$hours." jam yang lalu";
32	}
33	if(\$difftext==""){
34	if(\$minutes>=1)
35	\$difftext=\$minutes." menit yang lalu";
36	else
37	\$difftext="baru saja";
38	}
39	return \$difftext;
40	}
41	}

6.2.5 Implementasi Fungsi Memberikan Tanggapan

Fungsi memberikan tanggapan merupakan fungsi lanjutan dari fungsi memantau aspirasi, karena untuk memberikan tanggapan pengguna harus melewati fungsi tersebut terlebih dulu. Time Ago Helpers juga digunakan dalam kode algoritma *view* fungsi ini untuk menampilkan waktu tanggapan dikirimkan. Dalam Tabel 6.9 berikut ini merupakan hasil implementasi kode algoritma dari fungsi memberikan tanggapan yang terdapat dalam kelas *controller* Aspirasi.

Tabel 6.9 Implementasi Fungsi Mengirimkan Aspirasi

No	Kode Program
1	public function tambah_tanggapan(){
2	if (\$this->session->userdata('user_siap')) {
3	if (\$this->input->post('isi') != NULL) {
4	

Tabel 6.9 Implementasi Fungsi Mengirimkan Aspirasi (lanjutan)

No	Kode Program
5	<code>\$data = array('id_aspirasi' => \$this->input-</code>
6	<code>>post('id_aspirasi'),</code>
7	<code>'tanggapan' => \$this->input->post('isi'),</code>
8	<code>'username' => \$this->session->userdata('username'));</code>
9	<code>\$save = \$this->M_Aspirasi->addTanggapan(\$data);</code>
10	<code>\$tanggapan['tanggapan'] = \$this->M_Aspirasi-</code>
11	<code>>getTanggapan(array('id_aspirasi' => \$data['id_aspirasi']));</code>
12	<code>\$this->load->view('Aspirasi_Detail_Tanggapan',</code>
13	<code>\$tanggapan);</code>
14	<code>\$data_aspirasi = \$this->M_Aspirasi-</code>
15	<code>>getAspirasi(array('id_aspirasi' => \$data['id_aspirasi']))[0];</code>
16	<code>if (\$this->session->userdata('username') ==</code>
17	<code>\$data_aspirasi['username']) {</code>
18	<code>\$save = \$this->M_Aspirasi-</code>
19	<code>>addNotifikasi(array('id_aspirasi'=> \$data['id_aspirasi'],</code>
20	<code>'username' => 'admin'));</code>
21	<code>} else {</code>
22	<code>\$save = \$this->M_Aspirasi-</code>
23	<code>>addNotifikasi(array('id_aspirasi'=> \$data['id_aspirasi'],</code>
24	<code>'username' => \$data_aspirasi['username']));</code>
25	<code>\$data_pengirim = \$this->M_Aspirasi-</code>
26	<code>>getPengirim(array('username' => \$data_aspirasi['username']));</code>
27	<code>if (isset(\$data_pengirim['email'])) {</code>
28	<code>\$data_email = array('email' =></code>
29	<code>\$data_pengirim['email'],</code>
30	<code>'nama' => \$data_pengirim['nama'],</code>
31	<code>'id_aspirasi' => \$data['id_aspirasi']);</code>
32	<code>\$this-> kirim_notifikasi(\$data_email);</code>
33	<code>}</code>
34	<code>}</code>
35	<code>}</code>
36	<code>} else {</code>
37	<code>redirect(base_url(), 'refresh');</code>
38	<code>}</code>
39	<code>}</code>

6.2.6 Implementasi Fungsi Mengelola Akun

Aktor administrator dalam analisis persyaratan dapat mengelola akun yang terdaftar didalam sistem. Fitur yang dapat dilakukan administrator meliputi: membuat akun baru, edit dan hapus akun. Fitur tersebut hanya berlaku untuk data *user* yang bertipe admin (operator) dan lembaga, tidak berlaku untuk *user* yang bertipe mahasiswa karena data mahasiswa harus melewati API BAIS. Pada tabel 6.10 berikut ini merupakan hasil implementasi dari fungsi mengelola akun yang terdapat dalam kelas *controller* Admin.

Tabel 6.10 Implementasi Fungsi Mengelola Akun

No	Kode Program
1	<code>public function kelolaAkun(){</code>
2	<code>if (\$this->session->userdata('user_siap') AND (\$this->session-</code>
3	<code>>userdata('full_akses')==TRUE)) {</code>
4	<code>\$data['page'] = "Kelola Akun";</code>
5	<code>\$tipe = (\$this->uri->segment(3) ? htmlentities(\$this->uri-</code>
6	<code>>segment(3)): 'admin');</code>
7	<code>if (\$this->uri->segment(3) == 'edit') {</code>
8	<code>\$id = explode('-', \$this->uri->segment(4));</code>

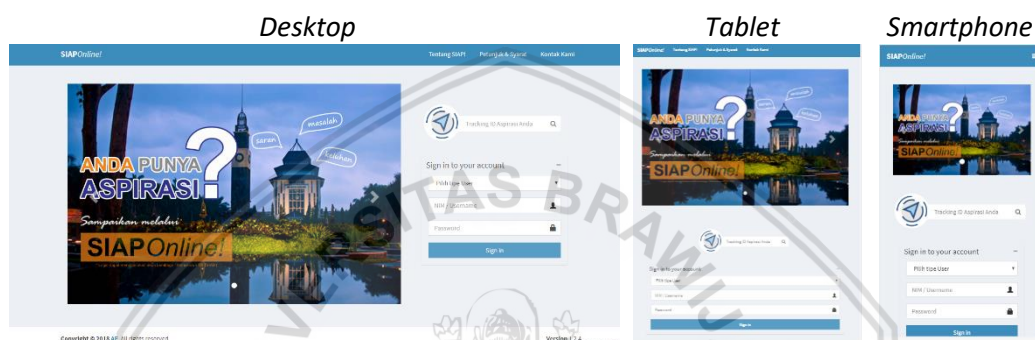
Tabel 6.10 Implementasi Fungsi Mengelola Akun (lanjutan)

No	Kode Program
9	if (substr(md5(\$id[1]), 0, 5) == \$this->uri->segment(5)) {
10	if (\$this->input->post('tipe') AND \$this->input-
11	>post('username')) {
12	\$dataUser = array('username' => \$this->input-
13	>post('username'),
14	'nama' => \$this->input->post('nama'),
15	'email' => \$this->input->post('email'),
16	'hp' => \$this->input->post('hp'),
17	'tipe' => \$id[0],
18	'verifikasi' => \$this->input->post('verifikasi'));
19	if (\$id[0] == 'admin') {
20	\$dataAdmin = array('username' => \$this->input-
21	>post('username'));
22	if (\$this->input->post('password2') != '' AND \$this-
23	>input->post('password2') == \$this->input->post('password1')) {
24	\$dataAdmin['password'] = md5(\$this->input-
25	>post('password2'));
26	}
27	\$save = \$this->M_User->updateAdmin(\$dataUser,
28	\$dataAdmin);
29	} elseif (\$id[0] == 'lembaga') {
30	\$dataL = array('username' => \$this->input-
31	>post('username'));
32	if (\$this->input->post('password2') != '' AND \$this-
33	>input->post('password2') == \$this->input->post('password1')) {
34	\$dataL['password'] = md5(\$this->input-
35	>post('password2'));
36	}
37	\$save = \$this->M_User->updateLembaga(\$dataUser,
38	\$dataL);
39	}
40	if (\$save) {
41	
42	redirect(base_url()."admin/kelolaAkun/\$id[0]","refresh");
43	}
44	} else {
45	\$data['akun'] = \$this->M_User->cekAkun(\$id[1])[0];
46	\$this->load->view('Admin_KelolaAkun_Edit', \$data);
47	}
48	}
49	} elseif (\$this->uri->segment(3) == 'hapus') {
50	\$id = explode('-', \$this->uri->segment(4));
51	if (substr(md5(\$id[1]), 0, 5) == \$this->uri->segment(5)) {
52	\$where = array('username' => \$id[1], 'tipe' => \$id[0]);
53	\$save = \$this->M_User->deleteAkun(\$where);
54	if (\$save) {
55	redirect(base_url()."admin/kelolaAkun/\$id[0]","refresh");
56	}
57	}
58	} else {
59	\$data['akun'] = \$this->M_User->getUsers(\$tipe);
60	\$this->load->view('Admin_KelolaAkun', \$data);
61	}
62	} else {
63	redirect(base_url(),"refresh");
64	}
65	}

6.3 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna akan menunjukkan beberapa hasil implementasi kedalam gambar antarmuka pengguna. Hal ini dilakukan berdasarkan hasil perancangan antarmuka yang telah dilakukan sebelumnya dalam penelitian ini. Semua hasil implementasi antarmuka pada sistem ini menggunakan *framework front-end* dari AdminLTE-3.0. Berikut ini merupakan beberapa hasil implementasi antarmuka pengguna menggunakan AdminLTE-3.0 yang telah dilakukan modifikasi agar sesuai dengan perancangan antarmuka pengguna dan analisis persyaratan yang telah ditentukan diawal penelitian.

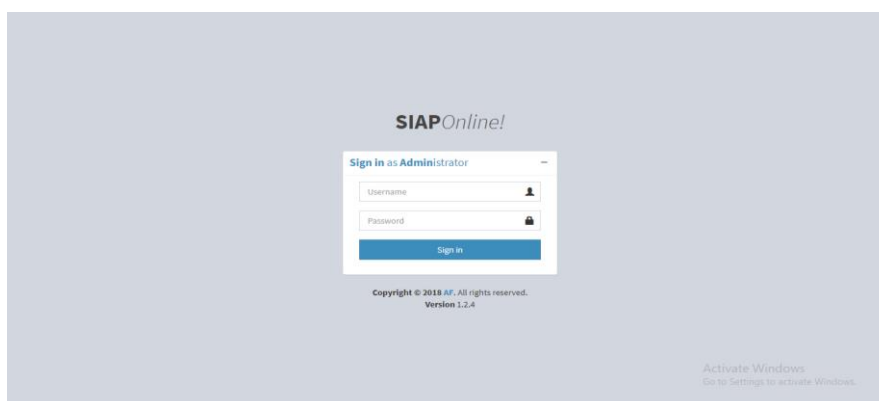
6.3.1 Implementasi Antarmuka Halaman Utama



Gambar 6.1 Antarmuka Halaman Utama

Gambar 6.1 menjelaskan bahwa dengan menerapkan konsep *responsive* maka tampilan SIAP Online DPM UB dapat menyesuaikan dengan web browser yang sedang mengaksesnya. Antarmuka Halaman utama ini terdiri dari menu navigasi, slider informasi/postingan, kolom pantau ID aspirasi yang dapat digunakan untuk memantau perkembangan aspirasi dan *form login* bagi mahasiswa atau lembaga yang belum login (*visitor*), *form login* tersebut terdiri dari kolom tipe, nim/username dan password. Sedangkan untuk operator dan administrator dalam implementasinya menggunakan antarmuka yang berbeda sebagaimana merujuk pada spesifikasi persyaratan awal sistem.

6.3.2 Implementasi Antarmuka Login Admin



Gambar 6.2 Antarmuka Login Admin

Untuk halaman antarmuka login Admin yang digambarkan pada Gambar 6.2, pada halaman tersebut yang hanya terdiri dari *form login* yang berisi kolom username dan password. Antarmuka login admin hanya dapat digunakan oleh operator dan administrator untuk masuk kedalam sistem.

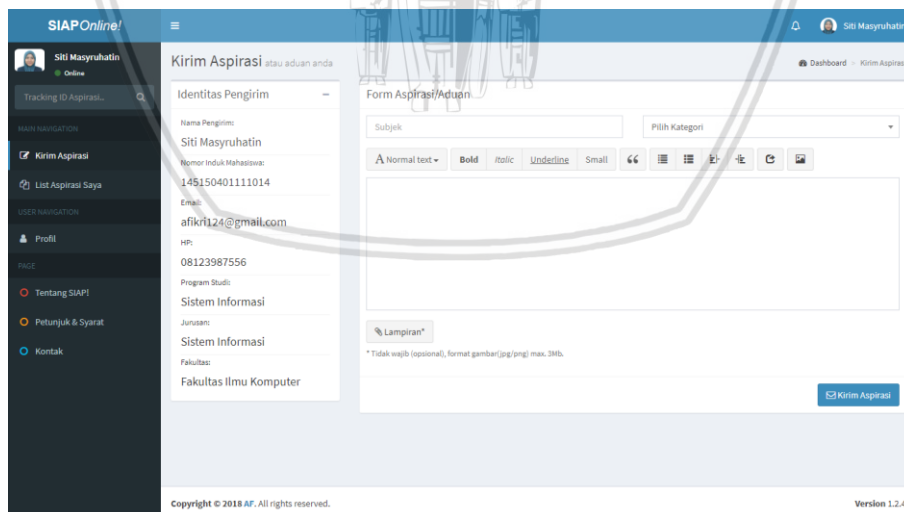
6.3.3 Implementasi Antarmuka Profil



Gambar 6.3 Antarmuka Profil

Pada Gambar 6.3 merupakan halaman antarmuka profil yang menampilkan identitas akun, pada halaman tersebut dapat memperbarui identitas dengan menekan tombol “Edit Profil”.

6.3.4 Implementasi Antarmuka Kirim Aspirasi



Gambar 6.4 Antarmuka Kirim Aspirasi

Pada Gambar 6.4 menunjukkan tampilan *form* Kirim aspirasi yang meminta masukan berupa subjek, kategori, isi aspirasi dan lampiran yang hanya menerima file dengan format gambar. Antarmuka ini hanya dapat digunakan oleh aktor mahasiswa dan lembaga.

6.3.5 Implementasi Antarmuka Statistik Aspirasi



Gambar 6.5 Antarmuka Statistik Aspirasi

Antarmuka statistik aspirasi yang digambarkan pada Gambar 6.5 merupakan halaman yang menampilkan total aspirasi, total pengirim, total tanggapan dan total aspirasi yang statusnya sudah selesai. Halaman ini juga menampilkan grafik batang dari jumlah aspirasi yang masuk perbulan yang ditampilkan berdasarkan tahun terpilih. Grafik donat menampilkan jumlah Aspirasi per fakultas juga berdasarkan tahun terpilih.

6.3.6 Implementasi Antarmuka Kelola Data Aspirasi

The screenshot displays the 'List Aspirasi' page in the SIAPOnline! system. It features a table of aspirations with columns for ID, Subjek, Pengirim, Kategori, Status, Ttp, Waktu, and Opsi. The table is filtered by 'Kategori' (Personal, Kelembagaan, Global) and 'Status' (Belum diproses, Dalam proses, Dalam tindak lanjut, Selesai). The table shows 10 rows of data, including details like 'ID', 'Subjek', 'Pengirim', 'Kategori', 'Status', 'Ttp', 'Waktu', and 'Opsi'.

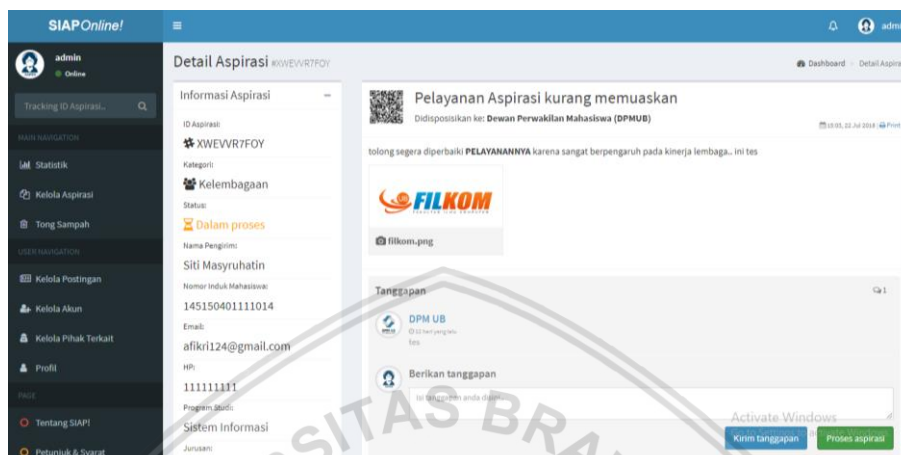
ID	Subjek	Pengirim	Kategori	Status	Ttp	Waktu	Opsi
127MCMY5AW	dana lembaga belum cair	14515040711074	Kelembagaan	Belum diproses	Q0	18 Jul 2018 07:43 (9 hari yang lalu)	[Icons]
PFT2PILAVC	DPM kurang mendengar	153140714111003	Personal	Belum diproses	Q0	15 Jul 2018 03:49 (9 hari yang lalu)	[Icons]
OHISTC3M6K	Sudah mengajukan Penurunan UKT, tapi belum turun	145150300111076	Personal	Belum diproses	Q1	11 Jul 2018 20:21 (12 hari yang lalu)	[Icons]
YLPKGEYXSF	Siapa pres DPM?	155150401111002	Personal	Belum diproses	Q0	11 Jul 2018 14:31 (12 hari yang lalu)	[Icons]
K53ZWU02H	Kurang Lahan Parkir	145150407111063	Personal	Dalam proses	Q0	11 Jul 2018 14:27 (12 hari yang lalu)	[Icons]
DL9TUIEICH	Makanan CL yang semakin mahal	145150401111059	Personal	Selesai (7/17/18 hari)	Q0	11 Jul 2018 14:15 (12 hari yang lalu)	[Icons]
LNP3SWKANB	Hilangnya harga diri Mahasiswa	145150401111068	Personal	Dalam proses	Q0	11 Jul 2018 14:12 (12 hari yang lalu)	[Icons]
XKONTESRFF	Dana UKM belum cair	145150407111074	Kelembagaan	Selesai (1 jam)	Q1	11 Jul 2018 00:12 (11 hari yang lalu)	[Icons]
LCYMEZY5B	UKT Mahal	145150407111074	Personal	Dalam tindak lanjut	Q0	11 Jul 2018 00:07 (11 hari yang lalu)	[Icons]
GKGMRYHNFH	Danasan yang memberatkan	145150400111001	Personal	Dalam proses	Q0	10 Jul 2018 23:09 (12 hari yang lalu)	[Icons]

Gambar 6.6 Antarmuka Kelola Data Aspirasi

Gambar 6.6 adalah tampilan antarmuka kelola data aspirasi yang masuk. Data aspirasi dapat dibagi berdasarkan kategori, diantaranya: personal, kelembagaan

dan global. Data aspirasi juga dapat dibagi berdasarkan status, diantaranya: belum diproses, dalam proses, dalam tindak lanjut dan selesai. Setiap aspirasi ditampilkan id aspirasi, subjek, pengirim, kategori, status, jumlah tanggapan, waktu aspirasi disampaikan dan opsi yang berisi tombol detail, cetak dan hapus.

6.3.7 Implementasi Antarmuka Detail Aspirasi



Gambar 6.7 Antarmuka Detail Aspirasi

Dengan menekan tombol detail di antarmuka kelola aspirasi atau melakukan *tracking* id aspirasi di halaman awal sistem, maka informasi detail aspirasi akan ditampilkan oleh sistem seperti pada Gambar 6.7. Pada halaman ini pengguna sistem dapat memberikan tanggapan, namun hanya operator dan administrator yang dapat memproses untuk merubah status aspirasi.

6.3.8 Implementasi Antarmuka Cetak Aspirasi

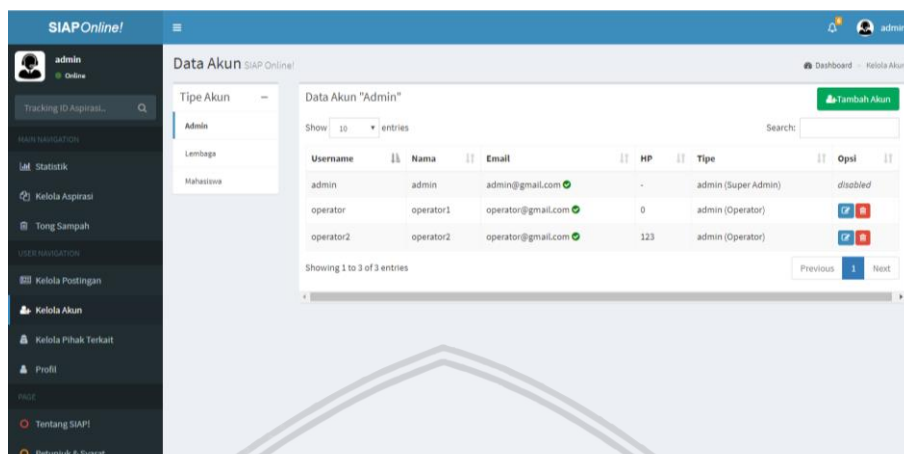


Gambar 6.8 Antarmuka Cetak Aspirasi

Ketika menekan fungsi cetak maka sistem akan menampilkan halaman cetak (Gambar 6.8) yang berisi detail aspirasi, meliputi: identitas pengirim, id aspirasi, subjek, tujuan/disposisi aspirasi kepada pihak terkait, waktu, status dan isi dari aspirasi tersebut. Pada tampilan ini juga menampilkan gambar qrcode yang dapat dipindai menggunakan kamera ponsel pintar untuk melihat detail dan

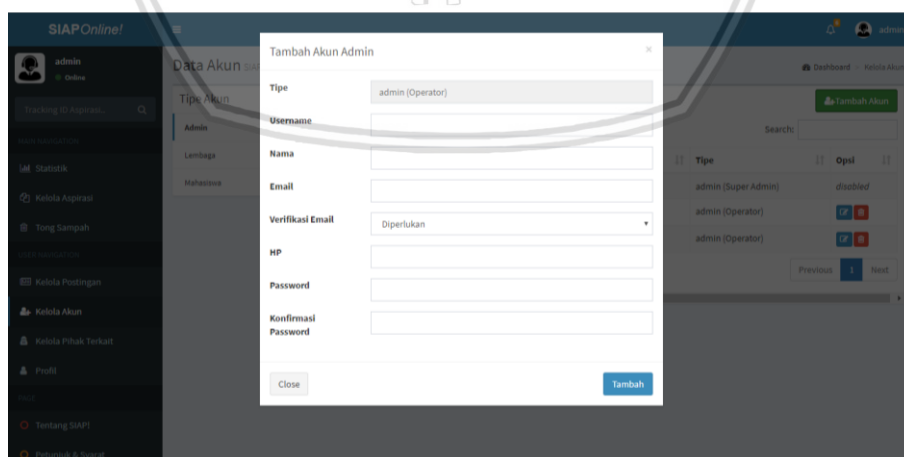
membuktikan keaslian dokumen cetak tersebut. Jika terdapat lampiran pada aspirasi tersebut maka akan di lampirkan pada halaman berikutnya (kedua).

6.3.9 Implementasi Antarmuka Kelola Akun



Gambar 6.9 Antarmuka Kelola Akun

Pada Gambar 6.9 antarmuka kelola akun, sistem akan menampilkan data-data akun berdasarkan tipe akun meliputi: admin (operator dan administrator), lembaga dan mahasiswa. Tabel data akun yang ditampilkan akan menyesuaikan dengan informasi yang dimiliki setiap tipe akun. Dari ketiga tipe akun untuk admin dan lembaga terdapat tombol tambah akun yang akan menampilkan pop-up seperti pada Gambar 6.10 yang dapat digunakan untuk menambahkan akun baru pada masing-masing kedua tipe tersebut. Tombol tambah akun tidak terdapat dalam antarmuka kelola akun dengan tipe mahasiswa, karena mahasiswa tidak perlu mendaftarkan diri atau didaftarkan, karena mahasiswa langsung dapat login menggunakan NIM dan password yang sama dengan yang digunakan login untuk SIAM.



Gambar 6.10 Antarmuka Kelola Akun – Tambah Akun

BAB 7 PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian yang dilakukan pada Sistem Informasi Aspirasi *Online* studi kasus pada DPM Universitas Brawijaya. Dengan menggunakan pemodelan *reuse-oriented development* maka pengujian yang dibutuhkan adalah dengan melakukan validasi sistem untuk menjamin bahwa sistem dapat berjalan sesuai dengan harapan. Berdasarkan rumusan masalah, pengujian dilakukan dengan menggunakan pengujian validasi untuk persyaratan fungsional sistem. Sedangkan pengujian kompatibilitas (*compatibility*), pengujian penerimaan pengguna (*user acceptance testing*) dan pengujian efisiensi waktu untuk menguji persyaratan non fungsional sistem.

7.1 Pengujian Validasi

Menurut Pressman (2010) validasi perangkat lunak yang menggunakan konteks OO (*object-oriented*) berfokus pada tindakan yang dapat dilihat pengguna dan keluaran yang dapat dikenali pengguna dari sistem. Pressman juga menambahkan bahwa untuk membantu dalam pengujian validasi, penguji harus memanfaatkan *use-case* yang merupakan bagian dari analisis persyaratan, karena *use-case* menyediakan skenario yang memiliki kemungkinan tinggi untuk menemukan kesalahan dalam persyaratan interaksi pengguna.

Tujuan dari pengujian rilis (yang di dalamnya terdapat pengujian validasi) adalah untuk memeriksa bahwa sistem sudah memenuhi persyaratannya dan cukup baik untuk digunakan pada lingkungan eksternal (Sommerville, 2011).

7.1.1 Rencana Pengujian Validasi

Untuk mendapatkan validasi sistem maka digunakan pengujian dengan metode *black-box* pada masing-masing *use-case* yang terdapat dalam persyaratan fungsional. Rencana pengujian validasi dilakukan menggunakan metode pengujian *black-box*, daftar kasus pengujiannya dapat dilihat pada Tabel 7.1 berikut ini.

Tabel 7.1 Rencana Pengujian Validasi

Kode Pengujian	Kode Use-Case	Nama Kasus Uji	Metode Pengujian
VAL-01	UC-01	Pengujian Login	<i>Black-box</i>
VAL-02	UC-02	Pengujian Logout	<i>Black-box</i>
VAL-03	UC-03	Pengujian Memperbarui Identitas	<i>Black-box</i>
VAL-04	UC-04	Pengujian Mengirimkan Aspirasi	<i>Black-box</i>
VAL-05	UC-05	Pengujian Mengelola Aspirasi	<i>Black-box</i>
VAL-06	UC-06	Pengujian Melihat Statistik Aspirasi	<i>Black-box</i>
VAL-07	UC-07	Pengujian Memberikan Tanggapan	<i>Black-box</i>

Tabel 7.1 Rencana Pengujian Validasi (lanjutan)

Kode Pengujian	Kode Use-Case	Nama Kasus Uji	Metode Pengujian
VAL-08	UC-08	Pengujian Memantau Aspirasi	<i>Black-box</i>
VAL-09	UC-09	Pengujian Mengelola Akun	<i>Black-box</i>
VAL-10	UC-10	Pengujian Menghapus Aspirasi	<i>Black-box</i>

7.1.2 Hasil Pengujian Validasi

Hasil pengujian validasi pada penelitian ini yang menggunakan metode pengujian *black-box* untuk membuktikan bahwa sistem telah memenuhi persyaratan yang diinginkan oleh pengguna. Berdasarkan rencana pengujian, pada setiap *use-case* dilakukan proses pengujian *black-box* yang masing-masing kasus uji akan ditunjukkan pada tabel 7.2 sampai 7.11.

Tabel 7.2 berikut ini menjelaskan hasil uji validasi pada kasus uji *login* sesuai dengan *use-case login* yang terdapat pada analisis persyaratan, kasus ini di uji menggunakan tipe pengguna sebagai mahasiswa.

Tabel 7.2 Kasus Uji Validasi Login

Kode pengujian	VAL-01
Kode use-case	UC-01
Kasus pengujian	Penguji melakukan <i>login</i> sebagai mahasiswa.
Tujuan pengujian	Pengujian untuk memastikan bahwa sistem dapat memberikan hak akses sesuai dengan identitas pengguna.
Prosedur pengujian	1. Penguji memilih tipe akun sebagai mahasiswa. 2. Penguji memasukkan NIM dan Password. 3. Penguji menekan tombol “masuk”.
Hasil yang diharapkan	Sistem memberikan hak akses sebagai mahasiswa.
Hasil yang didapatkan	Sistem memberikan hak akses sebagai mahasiswa.
Status pengujian	Valid

Tabel 7.3 berikut ini menjelaskan hasil uji validasi pada kasus uji *logout* sesuai dengan *use-case logout* yang terdapat pada analisis persyaratan, kasus uji ini dilakukan untuk memastikan bahwa sistem dapat mengakhiri session pengguna yang sedang *logged in*.

Tabel 7.3 Kasus Uji Validasi logout

Kode pengujian	VAL-02
Kode use-case	UC-02
Kasus pengujian	Penguji melakukan logout dari dalam sistem.
Tujuan pengujian	Pengujian untuk memastikan bahwa sistem dapat mengakhiri <i>session</i> pengguna yang sedang <i>logged in</i> .

Tabel 7.3 Kasus Uji Validasi logout (lanjutan)

Prosedur pengujian	1. Penguji memilih menu yang terdapat nama pengguna. 2. Penguji menekan tombol “keluar”.
Hasil yang diharapkan	Sistem berhasil mengakhiri <i>session</i> dan mengeluarkan pengguna dari dalam sistem.
Hasil yang didapatkan	Sistem berhasil mengakhiri <i>session</i> dan mengeluarkan pengguna dari dalam sistem.
Status pengujian	Valid

Tabel 7.4 berikut ini menjelaskan hasil uji validasi pada kasus uji memperbarui identitas sesuai dengan *use-case* memperbarui identitas yang terdapat pada analisis persyaratan, kasus uji ini bertujuan untuk memastikan bahwa sistem dapat mengganti *email* dan nomor *handphone* serta melakukan verifikasi *email*.

Tabel 7.4 Kasus Uji Validasi Memperbarui Identitas

Kode pengujian	VAL-03
Kode use-case	UC-03
Kasus pengujian	Penguji memperbarui <i>email</i> dan nomor <i>handphone</i> yang tersimpan di dalam sistem
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengganti <i>email</i> dan nomor <i>handphone</i> serta melakukan verifikasi pada <i>email</i> yang baru.
Prosedur pengujian	1. Penguji memilih menu “profil”. 2. Penguji menekan tombol “edit kontak”. 3. Penguji mengganti <i>email</i> yang lama dengan <i>email</i> yang baru. 4. Penguji mengganti nomor <i>handphone</i> yang lama dengan nomor <i>handphone</i> yang baru. 5. Penguji menekan tombol “simpan”. 6. Penguji menerima pesan <i>email</i> konfirmasi. 7. Penguji mengimputkan kode konfirmasi. 8. Penguji menekan tombol “konfirmasi”.
Hasil yang diharapkan	Sistem berhasil memperbarui identitas kontak <i>email</i> dan nomor <i>handphone</i> .
Hasil yang didapatkan	Sistem berhasil memperbarui identitas kontak <i>email</i> dan nomor <i>handphone</i> .
Status pengujian	Valid

Tabel 7.5 berikut ini menjelaskan hasil uji validasi pada kasus uji mengirimkan aspirasi sesuai dengan *use-case* mengirimkan aspirasi yang terdapat pada analisis persyaratan, kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menerima aspirasi yang disampaikan dan memberikan notifikasi pada pengguna.

Tabel 7.5 Kasus Uji Validasi Mengirimkan Aspirasi

Kode pengujian	VAL-04
Kode <i>use-case</i>	UC-04
Kasus pengujian	Penguji mengirimkan aspirasi dengan tambahan lampiran yang berupa gambar.
Tujuan pengujian	Memastikan bahwa sistem dapat menyimpan aspirasi serta lampirannya dan mengirimkan <i>email</i> notifikasi yang berisi ID aspirasi.
Prosedur pengujian	<ol style="list-style-type: none"> 1. Penguji memilih menu “Kirim Aspirasi”. 2. Penguji menginputkan subjek aspirasi. 3. Penguji memilih kategori aspirasi. 4. Penguji menginputkan pesan aspirasi. 5. Penguji menekan tombol “Lampiran” dan memilih gambar lampiran. 6. Penguji menekan tombol “Kirim Aspirasi”
Hasil yang diharapkan	Sistem berhasil menyimpan aspirasi dan mengirimkan <i>email</i> notifikasi.
Hasil yang didapatkan	Sistem berhasil menyimpan aspirasi dan mengirimkan <i>email</i> notifikasi.
Status pengujian	Valid

Tabel 7.6 berikut ini menjelaskan hasil uji validasi pada kasus uji mengelola aspirasi sesuai dengan *use-case* mengelola aspirasi yang terdapat pada analisis persyaratan, kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menampilkan data berdasarkan interaksi pengguna.

Tabel 7.6 Kasus Uji Validasi Mengelola Aspirasi

Kode pengujian	VAL-05
Kode <i>use-case</i>	UC-05
Kasus pengujian	Penguji mengklasifikasi data aspirasi berdasarkan status dan kategori dan penguji melakukan ekspor ke dalam <i>file excel</i> .
Tujuan pengujian	Memastikan bahwa sistem dapat menampilkan data berdasarkan klasifikasi yang dipilih dan mengekspornya ke dalam <i>file excel</i> .
Prosedur pengujian	<ol style="list-style-type: none"> 1. Penguji memilih menu “Kelola Aspirasi”. 2. Penguji memilih kategori dan status. 3. Penguji menekan tombol ekspor ke <i>excel</i>.
Hasil yang diharapkan	Sistem berhasil menampilkan klasifikasi data aspirasi berdasarkan kategori dan status yang dipilih, serta berhasil melakukan ekspor data kedalam <i>file excel</i> .
Hasil yang didapatkan	Sistem berhasil menampilkan klasifikasi data aspirasi berdasarkan kategori dan status yang dipilih, serta berhasil melakukan ekspor data kedalam <i>file excel</i>
Status pengujian	Valid

Tabel 7.7 berikut ini menjelaskan hasil uji validasi pada kasus uji melihat statistik aspirasi sesuai dengan *use-case* melihat statistik yang terdapat pada analisis persyaratan, kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menampilkan statistik dari aspirasi yang sudah tersimpan.

Tabel 7.7 Kasus Uji Validasi Melihat Statistik Aspirasi

Kode pengujian	VAL-06
Kode <i>use-case</i>	UC-06
Kasus pengujian	Penguji ingin melihat statistik aspirasi berdasarkan tahun tertentu.
Tujuan pengujian	Memastikan bahwa sistem dapat menampilkan statistik sesuai tahun yang dipilih.
Prosedur pengujian	1. Penguji memilih menu “Statistik”. 2. Penguji memilih tahun
Hasil yang diharapkan	Sistem berhasil menampilkan statistik aspirasi yang meliputi: total aspirasi, total pengirim, total tanggapan, total selesai, statistik bulanan, statistik fakultas, statistik waktu penyelesaian dan statistik waktu penyelesaian terakhir pada tahun terpilih.
Hasil yang didapatkan	Sistem berhasil menampilkan statistik aspirasi yang meliputi: total aspirasi, total pengirim, total tanggapan, total selesai, statistik bulanan, statistik fakultas, statistik waktu penyelesaian dan statistik waktu penyelesaian terakhir pada tahun terpilih.
Status pengujian	Valid

Tabel 7.8 berikut ini menjelaskan hasil uji validasi pada kasus uji memberikan tanggapan sesuai dengan *use-case* memberikan tanggapan yang terdapat pada analisis persyaratan, kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menerima tanggapan dari pengguna.

Tabel 7.8 Kasus Uji Validasi Memberikan Tanggapan

Kode pengujian	VAL-07
Kode <i>use-case</i>	UC-07
Kasus pengujian	Penguji memberikan tanggapan dan merubah status aspirasi.
Tujuan pengujian	Memastikan bahwa sistem dapat menyimpan tanggapan dan mengubah status aspirasi, mencetak (<i>print</i>) serta mengirimkan <i>email</i> notifikasi.
Prosedur pengujian	1. Penguji memilih menu “Kelola Aspirasi”. 2. Penguji menekan tombol lihat lebih detail. 3. Penguji mengimputkan tanggapannya. 4. Penguji menekan tombol “Kirim tanggapan”. 5. Penguji menekan tombol “Proses aspirasi” 6. Penguji memilih pihak yang terkait. 7. Penguji memilih status.

Tabel 7.8 Kasus Uji Validasi Memberikan Tanggapan (lanjutan)

	8. Penguji mencentang cetak (<i>print</i>) otomatis 9. Penguji menekan tombol “Simpan”
Hasil yang diharapkan	Sistem berhasil menyimpan tanggapan, berhasil mengubah status, berhasil menampilkan halaman cetak dan berhasil mengirimkan <i>email</i> notifikasi.
Hasil yang didapatkan	Sistem berhasil menyimpan tanggapan, berhasil mengubah status, berhasil menampilkan halaman cetak dan berhasil mengirimkan <i>email</i> notifikasi.
Status pengujian	Valid

Tabel 7.9 berikut ini menjelaskan hasil uji validasi pada kasus uji memantau aspirasi sesuai dengan *use-case* memantau aspirasi yang terdapat pada analisis persyaratan, kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menampilkan dapat digunakan untuk memantau aspirasi.

Tabel 7.9 Kasus Uji Validasi Memantau Aspirasi

Kode pengujian	VAL-08
Kode <i>use-case</i>	UC-08
Kasus pengujian	Penguji melakukan pantau aspirasi untuk melihat detail aspirasi.
Tujuan pengujian	Memastikan bahwa sistem dapat menampilkan detail aspirasi sesuai ID yang diinputkan.
Prosedur pengujian	1. Penguji menginputkan ID aspirasi pada <i>form</i> pantau. 2. Penguji menekan tombol enter pada keyboard.
Hasil yang diharapkan	Sistem berhasil menampilkan detail aspirasi sesuai ID yang diinputkan.
Hasil yang didapatkan	Sistem berhasil menampilkan detail aspirasi sesuai ID yang diinputkan.
Status pengujian	Valid

Tabel 7.10 berikut ini menjelaskan hasil uji validasi pada kasus uji mengelola akun sesuai dengan *use-case* mengelola akun yang terdapat pada analisis persyaratan, kasus uji ini dilakukan untuk memastikan bahwa sistem dapat mengelola akun pengguna sistem.

Tabel 7.10 Kasus Uji Validasi Mengelola Akun

Kode pengujian	VAL-09
Kode <i>use-case</i>	UC-09
Kasus pengujian	Penguji membuat akun baru lalu menghapus akun tersebut.
Tujuan pengujian	Memastikan bahwa sistem dapat menambahkan akun baru dan juga dapat menghapus akun yang sudah terdaftar.
Prosedur pengujian	1. Penguji memilih menu “Kelola akun”. 2. Penguji memilih tipe akun “Lembaga”.

Tabel 7.10 Kasus Uji Validasi Mengelola Akun (lanjutan)

	3. Penguji menekan tombol “Tambah Akun”. 4. Penguji menginputkan <i>username</i> , nama, <i>email</i> , <i>password</i> dan konfirmasi <i>password</i> . Lalu menekan tombol “Tambah”. 5. Penguji menekan tombol hapus <i>user</i> pada baris akun yang baru saja dibuat.
Hasil yang diharapkan	Sistem berhasil menambahkan akun baru dan juga berhasil menghapusnya.
Hasil yang didapatkan	Sistem berhasil menambahkan akun baru dan juga berhasil menghapusnya.
Status pengujian	Valid

Tabel 7.11 berikut ini menjelaskan hasil uji validasi pada kasus uji menghapus aspirasi sesuai dengan *use-case* menghapus aspirasi yang terdapat pada analisis persyaratan, kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menghapus aspirasi yang tersimpan.

Tabel 7.11 Kasus Uji Validasi Menghapus Aspirasi

Kode pengujian	VAL-10
Kode <i>use-case</i>	UC-10
Kasus pengujian	Penguji menghapus aspirasi tertentu lalu melihat aspirasi tersebut di menu tong sampah.
Tujuan pengujian	Memastikan bahwa sistem dapat menghapus aspirasi dari daftar kelola aspirasi dan memindahkannya ke tong sampah.
Prosedur pengujian	1. Penguji memilih menu “Kelola aspirasi”. 2. Penguji menekan tombol hapus aspirasi ini. 3. Penguji mengkonfirmasi penghapusan aspirasi.
Hasil yang diharapkan	Sistem berhasil menghilangkan aspirasi yang dihapus di tabel kelola aspirasi dan menampilkannya di tabel tong sampah.
Hasil yang didapatkan	Sistem berhasil menghilangkan aspirasi yang dihapus di tabel kelola aspirasi dan menampilkannya di tabel tong sampah.
Status pengujian	Valid

7.1.3 Analisis Hasil Pengujian Validasi

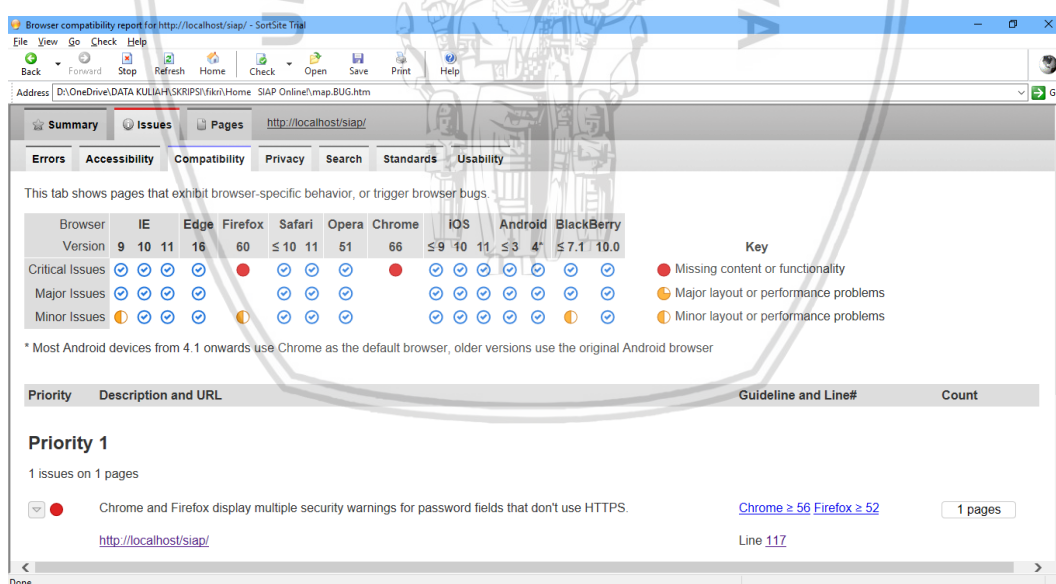
Hasil analisis pengujian validasi sistem dengan menggunakan metode pengujian *black-box* dilakukan untuk mengetahui validasi dari masing-masing kesesuaian antara persyaratan sistem dengan kinerja sistem saat dijalankan. Berdasarkan hasil pengujian pada sub-bab 7.1.2 sebelumnya, maka didapatkan kesimpulan bahwa semua persyaratan fungsional sistem yang telah dilakukan pengujian memiliki hasil yang sesuai antara kinerja sistem dengan persyaratan sistem yang telah ditentukan pada tahapan analisis persyaratan, dengan kata lain sistem ini memiliki hasil uji validasi dengan status 100% valid karena mendapatkan 10 nilai valid dari 10 pengujian yang dilakukan.

7.2 Pengujian Kompatibilitas

Seperti yang telah dijelaskan dalam bab landasan kepustakaan tentang pengujian kompatibilitas, pengujian ini berfokus terhadap persyaratan non-fungsional yang bertujuan untuk memastikan sistem yang telah dibuat mampu berjalan dengan baik pada lingkungan yang berbeda, sehingga hasil dari pengujian ini dapat menyatakan kemudahan dalam mengakses sistem yang telah dikembangkan. Pengujian kompatibilitas ini dilakukan menggunakan alat bantu *software SortSite* yang memiliki tujuan untuk menguji *compatibility* dan *accessibility* sistem pada beberapa *browser* dengan beberapa versi dan jenis jenis *browser* yang ada.

7.2.1 Hasil Pengujian Kompatibilitas

Hasil pengujian kompatibilitas pada Gambar 7.1 merupakan hasil dari pengujian dengan alat bantu *software SortSite*, dapat dilihat pada gambar tersebut terdapat isu kritis pada *browser Firefox* (versi 60) dan *Chrome* (versi 66), isu tersebut karena terdapat *form login* yang memiliki tipe input berupa *password*. *Software SortSite* menyarankan untuk menggunakan protokol HTTPS (*Hypertext Transfer Protocol Secure*) sedangkan sistem diuji di *localhost* yang hanya menggunakan protokol HTTP (*Hypertext Transfer Protocol*). Dalam hasil pengujian juga terdapat empat isu minor yang menyatakan bahwa terdapat CSS (*Cascading Style Sheet*) yang tidak *support* untuk beberapa *browser*.



Gambar 7.1 Hasil Pengujian Kompatibilitas

7.2.2 Analisis Hasil Pengujian Kompatibilitas

Dari hasil pengujian kompatibilitas yang telah dilakukan pada tahapan sebelumnya, ditemukan permasalahan kritis pada hasil uji menggunakan *browser Firefox* (versi 60) dan *Chrome* (versi 66), solusi dari permasalahan tersebut dapat terselesaikan jika sistem sudah diunggah ke hosting dengan domain yang menggunakan protokol HTTPS. Dalam hasil pengujian tersebut juga terdapat

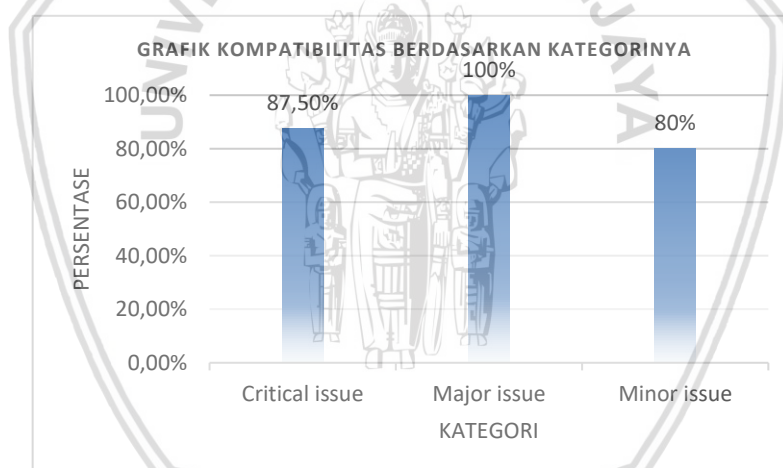
sedikit isu *minor* pada *Internet Explorer* versi 9, *Firefox* versi 60 dan *BlackBerry* versi 7.1 ke bawah, setelah dilakukan pengecekan permasalahannya terdapat pada sebagian CSS yang mengatur tampilan namun dengan masalah tersebut sistem tetap dapat berjalan sesuai persyaratan pengguna.

Dapat dilihat pada Gambar 7.1 yang terdapat banyak tanda centang, hasil tersebut menunjukkan bahwa hasil uji kompatibilitas sistem berdasarkan kategorinya didapatkan kesimpulan pada Tabel 7.12 berikut ini.

Tabel 7.12 Hasil Uji Kompatibilitas

Kategori	Tingkat Kompatibilitas
<i>Critical issue</i>	14 Berhasil dari 16 Browser
<i>Major issue</i>	14 Berhasil dari 14 Browser
<i>Minor issue</i>	12 Berhasil dari 15 Browser

Hal ini menandakan bahwa tingkat kompatibilitas sistem yang telah diuji berjalan dengan baik karena mendapatkan skor rata-rata 89% seperti yang digambarkan pada Gambar 7.2 berikut ini.



Gambar 7.2 Grafik Persentase Hasil Uji Kompatibilitas

7.3 Pengujian Penerimaan Pengguna

Pengujian penerimaan pengguna (*user acceptance testing*) dilakukan untuk mendapatkan konfirmasi bahwa suatu sistem sudah memenuhi persyaratan yang disepakati bersama. Dilakukan verifikasi akhir menggunakan pengujian penerimaan pengguna agar mengetahui apakah fungsi bisnis sudah sesuai dengan yang diperlukan dan agar mengetahui ketepatan fungsi sistem dengan meniru kondisi penggunaan sebenarnya (Lau, 2012).

Maka dari itu perlu untuk melakukan skenario pengujian pada masing-masing tipe pengguna sistem, yaitu: mahasiswa, lembaga, dan anggota DPM UB selaku operator dan administrator sistem. Skenario pengujian dilakukan berdasarkan kebutuhan sebenarnya dari masing-masing pengguna.

Tahapan pertama adalah mendefinisikan kriteria penerimaan, namun karena mengacu pada *template* pertanyaan UAT yang sudah tersedia, tahapan ini dapat dilewati karena dalam *template* tersebut terdapat dua jenis kriteria pertanyaan yaitu pertanyaan umum (*usability* dan *security*) serta pertanyaan berdasarkan fungsinya. Selanjutnya tahapan kedua menentukan sumberdaya, waktu, biaya dan jadwal pengujian. Menurut Hendryadi & Suryani (2015) ukuran sampel penelitian eksperimental sederhana yaitu sebanyak 10-20 sampel dengan kontrol yang ketat atau 10% dari populasi untuk kontrol yang tidak ketat. Dari total 60 ribuan mahasiswa UB peneliti menggunakan 15 orang mahasiswa sebagai responden dengan kontrol yang ketat dan menggunakan 10% dari total lembaga dan anggota DPM UB. Tahapan ketiga membuat desain pengujian yaitu dengan menggunakan skenario tugas yang harus dilakukan responden. Skenario tersebut terdapat dalam Tabel 7.13 sampai Tabel 7.15 berikut ini.

Tabel 7.13 Skenario UAT pada Mahasiswa/Lembaga

No	Tugas yang dikerjakan
1	Login sistem menggunakan NIM dan password SIAM/akun Lembaga
2	Menekan tombol “Kirim Aspirasi”
3	Mengisi form aspirasi
4	Menekan tombol “Kirim Aspirasi”
5	Mengecek email notifikasi yang terdapat ID aspirasi
6	Memasukkan ID aspirasi pada form pantau aspirasi
7	Mengisi form tanggapan
8	Menekan tombol “Kirim Tanggapan”
9	Pergi ke menu “List Apirasi Saya” untuk melihat aspirasi yang pernah dikirimkan

Tabel 7.14 Skenario UAT pada Operator

No	Tugas yang dikerjakan
1	Melakukan login sistem menggunakan akun operator
2	Menekan menu “Kelola Aspirasi”
3	Menekan tombol “Lihat lebih detail” pada aspirasi yang statusnya masih belum diproses
4	Mengisi Form tanggapan
5	Menekan tombol “Kirim tanggapan”
6	Menekan tombol “Proses Aspirasi”
7	Memilih tujuan disposisi, mengubah status, mencetak aspirasi dan menekan tombol “Simpan”
8	Pergi ke menu “Statistik” untuk melihat statistik aspirasi
9	Memilih tahun untuk menampilkan data statistik berdasarkan tahun tertentu

Tabel 7.15 Skenario UAT pada Administrator

No	Tugas yang dikerjakan
1	Melakukan login sistem menggunakan akun Administrator
2	Menekan menu “Kelola Aspirasi”
3	Menekan tombol “Lihat lebih detail” pada aspirasi yang statusnya masih belum diproses
4	Mengisi Form tanggapan
5	Menekan tombol “Kirim tanggapan”
6	Menekan tombol “Proses Aspirasi”
7	Memilih tujuan disposisi, mengubah status, mencetak aspirasi dan menekan tombol “Simpan”
8	Menekan kembali menu “Kelola Aspirasi”
9	Menekan tombol “Ekspor ke Excel” untuk mengekspor data ke dalam format excel
10	Menekan tombol “Hapus Aspirasi” pada data yang teridentifikasi spam
11	Pergi ke menu “Statistik” untuk melihat statistik aspirasi
12	Memilih tahun untuk menampilkan data statistik berdasarkan tahun tertentu
13	Menekan menu “Kelola Akun”
14	Menekan tombol “Tambah Akun”
15	Mangisi form tambah akun
16	Menekan tombol “Tambah”
17	Menekan tombol “Hapus User” pada user yang ingin dihapus

Tahapan keempat adalah menjalankan pengujian penerimaan (*run acceptance test*) dengan menemui responden secara langsung, peneliti menjelaskan mengenai sistem yang sedang diuji dan memberikan gambaran bagaimana sistem tersebut bekerja kepada responden, setelah itu responden diperkenalkan untuk mencoba sistem sesuai skenario pengguna yang telah dibuat sebelumnya. Selanjutnya responden diminta untuk mengisi kuesioner untuk mengetahui reaksi pengguna terhadap sistem dan mengetahui apakah fungsional sistem sudah dapat diterima oleh pengguna akhir. Kuesioner yang diberikan pada masing-masing responden terdapat dua jenis yang terdiri dari pernyataan secara umum dan pernyataan berdasarkan fungsi sistem dari masing-masing tipe pengguna. Contoh kuesioner yang digunakan dapat dilihat pada Lampiran C kuesioner pengujian penerimaan pengguna. Dari hasil kuesioner tersebut dapat disimpulkan hasil dari uji penerimaan pengguna berdasarkan perhitungan skala *likert*.

7.3.1 Hasil Pengujian Penerimaan Pengguna

Berdasarkan hasil kuesioner yang dilakukan pada total 26 responden yang terdiri dari 15 orang mahasiswa, 5 orang perwakilan lembaga dan 6 orang anggota DPM UB, maka didapatkan hasil pada Tabel 7.16. Total responden tersebut sudah melebihi dari minimum ukuran sampel penelitian eksperimental, penentuan jumlah kategori sampel perwakilan lembaga dan anggota DPM UB mengambil 10% dari populasi yang ada, dari total LKM yang ada di UB sebanyak 47 lembaga dan anggota DPM UB yang totalnya 65 orang maka dapat ditentukan 10% dari sampel responden yang dibutuhkan adalah sebanyak 5 responden perwakilan lembaga dan 6 responden perwakilan anggota DPM UB.

Tabel 7.16 Hasil Kuesioner Pengujian Penerimaan Pengguna

No	Pernyataan	Jumlah Nilai					Jumlah
		STS	TS	N	S	SS	
Pernyataan Umum							
1	Saya mendapatkan antarmuka sistem yang mudah untuk digunakan.		1	3	16	6	26
2	Saya merasa aman dan nyaman dalam berinteraksi dengan sistem	1	1	6	12	6	26
3	Saya mendapatkan respon sistem yang konsisten dan dapat diprediksi			6	13	7	26
4	Nilai sistem berdasarkan frasa berikut ini:						
	a. Informasi sangat mudah ditemukan		1	3	11	11	26
	b. Jumlah gambar/grafiknya sangat sesuai		1	8	11	6	26
	c. Kegunaannya sangat jelas dan tepat		3	2	7	14	26
	d. Saya dapat mengetahui bagaimana cara mendapatkan informasi yang saya inginkan	1	1	9	11	4	26
	e. Jumlah penjelasan cukup untuk melakukan tugas		1	9	13	3	26
Total		2	9	46	94	57	208
Pernyataan Berdasarkan Fungsinya							
1	Tipe Mahasiswa/Lembaga						
	a. Fungsi yang terdapat dalam sistem memungkinkan saya untuk menyampaikan dan memantau aspirasi saya secara lebih efektif dan efisien			5	8	7	20
	b. Sistem memberikan saya informasi yang saya perlukan secara ringkas		1	5	9	5	20
	c. Saya mendapatkan sistem yang berguna dalam pelayanan aspirasi		2	2	9	7	20
2	Tipe Operator dan Administrator						

Tabel 7.16 Hasil Kuesioner Pengujian Penerimaan Pengguna (lanjutan)

No	Pernyataan	Jumlah Nilai					Jumlah
		STS	TS	N	S	SS	
	a. Fungsi yang terdapat dalam sistem memungkinkan saya untuk mengelola aspirasi secara lebih efektif dan efisien				4	2	6
	b. Sistem dapat melakukan tugas pelayanan aspirasi berdasarkan instruksi saya				5	1	6
	c. Tampilan statistik memungkinkan saya untuk mengambil keputusan tentang bagaimana tugas yang harus saya lakukan			1	2	3	6
	d. Tampilan statistik dapat membantu saya dalam memperhatikan penanganan aspirasi yang perlu disegerakan				3	3	6
3	Tipe Administrator						
	a. Sistem ini membantu saya mengubah perencanaan/tindakan dalam menangani aspirasi				1		1
	b. Sistem ini membantu saya untuk mematuhi standar pelayanan terbaik dalam menangani aspirasi mahasiswa				1		1
	c. Sistem ini memungkinkan saya untuk mengambil keputusan yang baik tentang bagaimana tindakan yang harus saya lakukan				1		1
Total		0	3	13	43	28	87

Keterangan:

STS : Sangat Tidak Setuju

S : Setuju

TS : Tidak Setuju

SS : Sangat Setuju

N : Netral

7.3.2 Analisis Hasil Pengujian Penerimaan Pengguna

Untuk mengetahui hasil tanggapan dari responden (pengguna) terhadap Sistem Informasi Aspirasi *Online* studi kasus pada DPM UB, maka selanjutnya adalah melakukan analisis terhadap hasil kuesioner *user acceptance testing* (UAT) yang dilakukan pada 26 responden, dimana jawaban dari pertanyaan tersebut terdiri dari tingkatan yang dapat dipilih dengan pembobotan masing-masing jawabannya terdapat pada Tabel 7.17 berikut:

Tabel 7.17 Bobot Nilai Pilihan Jawaban UAT

Jawaban		Bobot
SS	Sangat Setuju	5
S	Setuju	4
N	Netral	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

Selanjutnya, hasil yang didapatkan pada Tabel 7.16 diolah menggunakan rumus skala *likert* yaitu mengkalikan hasil dengan bobot yang sudah ditentukan sesuai Tabel 7.17. Dari hasil perhitungan dengan mengkalikan setiap jawaban bobot yang sudah ditentukan, maka didapatkan hasil dalam Tabel 7.18 berikut ini.

Tabel 7.18 Hasil Kuesioner UAT Setelah Diolah

No	Pernyataan	Jumlah Nilai					Jumlah x bobot
		STS	TS	N	S	SS	
		x 1	x 2	x 3	x 4	x 5	
Pernyataan Umum							
1	Saya mendapatkan antarmuka sistem yang mudah untuk digunakan.	0	2	9	64	30	105
2	Saya merasa aman dan nyaman dalam berinteraksi dengan sistem	1	2	18	48	30	99
3	Saya mendapatkan respon sistem yang konsisten dan dapat diprediksi	0	0	18	52	35	105
4	Nilai sistem berdasarkan frasa berikut ini:						
	a. Informasi sangat mudah ditemukan	0	2	9	44	55	110
	b. Jumlah gambar/grafiknya sangat sesuai	0	2	24	44	30	100
	c. Kegunaannya sangat jelas dan tepat	0	6	6	28	70	110
	d. Saya dapat mengetahui bagaimana cara mendapatkan informasi yang saya inginkan	1	2	27	44	20	94
	e. Jumlah penjelasan cukup untuk melakukan tugas	0	2	27	52	15	96
Total		2	2	18	138	376	285
Pernyataan Berdasarkan Fungsinya							
1	Tipe Mahasiswa/Lembaga						
	a. Fungsi yang terdapat dalam sistem memungkinkan saya untuk menyampaikan dan memantau aspirasi saya secara lebih efektif dan efisien	0	0	15	32	35	82
	b. Sistem memberikan saya informasi yang saya perlukan secara ringkas	0	2	15	36	25	78

Tabel 7.18 Hasil Kuesioner UAT Setelah Diolah (lanjutan)

No	Pernyataan	Jumlah Nilai					Jumlah x bobot
		STS x 1	TS x 2	N x 3	S x 4	SS x 5	
	c. Saya mendapatkan sistem yang berguna dalam pelayanan aspirasi	0	4	6	36	35	81
2	Tipe Operator dan Administrator						
	a. Fungsi yang terdapat dalam sistem memungkinkan saya untuk mengelola aspirasi secara lebih efektif dan efisien	0	0	0	16	10	26
	b. Sistem dapat melakukan tugas pelayanan aspirasi berdasarkan instruksi saya	0	0	0	20	5	25
	c. Tampilan statistik memungkinkan saya untuk mengambil keputusan tentang bagaimana tugas yang harus saya lakukan	0	0	3	8	15	26
	d. Tampilan statistik dapat membantu saya dalam memperhatikan penanganan aspirasi yang perlu disegerakan	0	0	0	12	15	27
3	Tipe Administrator						
	a. Sistem ini membantu saya mengubah perencanaan/tindakan dalam menangani aspirasi	0	0	0	4	0	4
	b. Sistem ini membantu saya untuk mematuhi standar pelayanan terbaik dalam menangani aspirasi mahasiswa	0	0	0	4	0	4
	c. Sistem ini memungkinkan saya untuk mengambil keputusan yang baik tentang bagaimana tindakan yang harus saya lakukan	0	0	0	4	0	4
Total		0	0	6	39	172	140

Setelah mengkalikan nilai jawaban dengan bobot dari masing-masing jawaban, dapat dilakukan analisa dari kolom jumlah x bobot pada tabel setelah diolah dengan jumlah responden dari masing-masing pertanyaan. Analisa dapat dilakukan dengan cara membagi jumlah data yang sudah diolah dengan jumlah responden masing-masing pertanyaan untuk mendapatkan rata-ratanya, lalu untuk mendapatkan persentase nilainya, hasil rata-rata dibagi dengan jumlah jawaban yaitu 5 lalu dikalikan 100. Hasil analisisnya dijabarkan dalam Tabel 7.19 berikut ini.

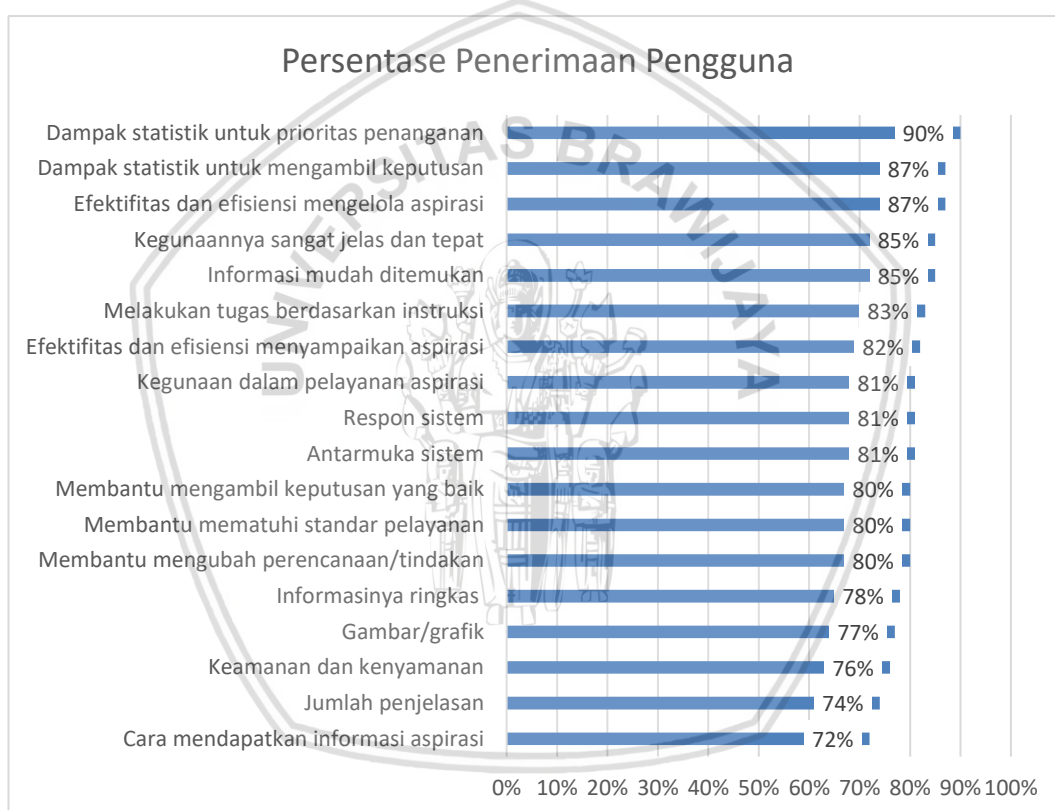
Tabel 7.19 Hasil Analisa Kuesioner UAT

No	Pernyataan	Jumlah sampel	Jumlah x bobot	Rata-rata	Persentase
1	Saya mendapatkan antarmuka sistem yang mudah untuk digunakan.	26	105	4,0	81%
2	Saya merasa aman dan nyaman dalam berinteraksi dengan sistem	26	99	3,8	76%
3	Saya mendapatkan respon sistem yang konsisten dan dapat diprediksi	26	105	4,0	81%
4	Informasi sangat mudah ditemukan	26	110	4,2	85%
5	Jumlah gambar/grafiknya sangat sesuai	26	100	3,8	77%
6	Kegunaannya sangat jelas dan tepat	26	110	4,2	85%
7	Saya dapat mengetahui bagaimana cara mendapatkan informasi yang saya inginkan	26	94	3,6	72%
8	Jumlah penjelasan cukup untuk melakukan tugas	26	96	3,7	74%
9	Fungsi yang terdapat dalam sistem memungkinkan saya untuk menyampaikan dan memantau aspirasi saya secara lebih efektif dan efisien	20	82	4,1	82%
10	Sistem memberikan saya informasi yang saya perlukan secara ringkas	20	78	3,9	78%
11	Saya mendapatkan sistem yang berguna dalam pelayanan aspirasi	20	81	4,1	81%
12	Fungsi yang terdapat dalam sistem memungkinkan saya untuk mengelola aspirasi secara lebih efektif dan efisien	6	26	4,3	87%
13	Sistem dapat melakukan tugas pelayanan aspirasi berdasarkan instruksi saya	6	25	4,2	83%
14	Tampilan statistik memungkinkan saya untuk mengambil keputusan tentang bagaimana tugas yang harus saya lakukan	6	26	4,3	87%
15	Tampilan statistik dapat membantu saya dalam memperhatikan penanganan aspirasi yang perlu disegerakan	6	27	4,5	90%
16	Sistem ini membantu saya mengubah perencanaan/tindakan dalam menangani aspirasi	1	4	4,0	80%
17	Sistem ini membantu saya untuk mematuhi standar pelayanan terbaik dalam menangani aspirasi mahasiswa	1	4	4,0	80%
18	Sistem ini memungkinkan saya untuk mengambil keputusan yang baik tentang	1	4	4,0	80%

Tabel 7.19 Hasil Analisa Kuesioner UAT (lanjutan)

No	Pernyataan	Jumlah sampel	Jumlah x bobot	Rata-rata	Persentase
	bagaimana tindakan yang harus saya lakukan				
Rata-rata keseluruhan					81%

Dari data dalam Tabel 7.19 maka dapat dijadikan grafik hasil uji penerimaan pengguna terhadap Sistem Informasi Aspirasi *Online* studi kasus pada DPM UB seperti yang digambarkan dalam Gambar 7.3 berikut ini.



Gambar 7.3 Grafik Persentase Hasil Uji Penerimaan Pengguna

Berdasarkan Gambar 7.3 maka hasil uji penerimaan pengguna dapat diambil kesimpulan bahwa sistem sudah dapat diterima oleh pengguna akhir, karena mendapatkan nilai rata-rata keseluruhan penerimaan pengguna sebesar 81% yang berarti sangat setuju untuk digunakan dalam menangani pelayanan aspirasi pada DPM UB.

7.4 Pengujian Efisiensi Waktu

Pengujian ini dilakukan untuk membandingkan waktu yang dibutuhkan dalam menjalankan aktivitas dalam proses bisnis saat ini (*as-is*) dengan proses bisnis usulan (*to-be*). Skenario simulasi pengujian ini dilakukan pada mahasiswa dan lembaga selaku pengaspirasi serta anggota DPM UB selaku penyerap aspirasi dan pengelolanya. Simulasi dilakukan pada 5 orang responden mahasiswa atau lembaga dan 2 orang responden dari anggota komisi III DPM UB untuk melakukan aktivitas yang terdapat dalam skenario pemodelan proses bisnis *as-is* dan *to-be* pada masing-masing tipe pengguna tersebut.

7.4.1 Hasil Pengujian Efisiensi Waktu

Hasil uji yang dilakukan menggunakan simulasi skenario yang telah ditentukan sebelumnya mendapatkan hasil perhitungan waktu dalam menjalankan setiap aktivitas pada masing-masing proses bisnis. Aktivitas yang dilakukan mahasiswa/lembaga dalam skenario merupakan aktivitas dari sebelum menyampaikan aspirasi hingga mendapatkan hasil dari aspirasi yang disampaikan berdasarkan model proses bisnis *as-is* melalui sekretariat DPM UB. Sedangkan untuk anggota komisi III DPM UB merupakan aktivitas yang dilakukan dalam model proses bisnis *as-is* melalui kegiatan safari fakultas.

Hasil perhitungan waktu yang didapatkan dari masing-masing model bisnis pada mahasiswa atau lembaga terdapat dalam Tabel 7.20. Sedangkan hasil perhitungan waktu yang didapatkan dari masing-masing model bisnis pada anggota komisi III DPM UB terdapat dalam Tabel 7.21. Dari hasil tersebut selanjutnya dilakukan analisis pada tahapan selanjutnya untuk mendapatkan efisiensi dari perbandingan model proses bisnis yang telah dilakukan pengujian.

Dalam Tabel 7.20 yang berdasarkan pemodelan proses bisnis dalam Bab 4, pemodelan proses bisnis *as-is* yang dimulai dari mahasiswa mendatangi sekretariat DPM UB, lalu menyampaikan keperluannya pada anggota DPM UB yang berada di sekretariat, jika anggota komisi III berada di sekretariat maka mahasiswa/lembaga dapat menemui anggota komisi III lalu menyampaikan aspirasinya, setelah itu tinggal menunggu aspirasi diproses dan mahasiswa akan dihubungi anggota DPM UB untuk mendapatkan hasilnya. Sedangkan pada model proses bisnis *to-be*, dimulai dari mahasiswa/lembaga yang mengakses Sistem Informasi Aspirasi *Online* (SIAP *Online*) DPM UB dengan melakukan login sebagai mahasiswa/lembaga sesuai akun masing-masing. Setelah itu, mahasiswa/lembaga dapat menyampaikan melalui form aspirasi yang disediakan oleh sistem. Mahasiswa/lembaga akan mendapatkan notifikasi email yang berisi ID aspirasi yang dapat digunakan untuk memantau perkembangan aspirasi atau melalui dashboard sistem yang telah disediakan, yang juga dapat untuk memberikan tanggapan terkait perkembangan aspirasinya. Setelah aspirasi diproses mahasiswa/lembaga akan mendapatkan notifikasi email bahwa aspirasinya sudah mendapatkan hasil. Sehingga mahasiswa/lembaga cukup memantau menggunakan sistem untuk mendapatkan hasilnya.

Tabel 7.20 Hasil perhitungan waktu yang dibutuhkan mahasiswa dan lembaga

No	aktivitas yang dikerjakan		Waktu yang diperlukan									
			Responden 1		Responden 2		Responden 3		Responden 4		Responden 5	
	As-Is	To-Be	As-Is	To-Be	As-Is	To-Be	As-Is	To-Be	As-Is	To-Be	As-Is	To-Be
1	Mendatangi sekretariat DPM UB	Mengakses SIAP Online DPM UB	00.30.00	00.00.10	00.15.00	00.00.15	00.20.00	00.00.11	00.10.00	00.00.10	00.10.00	00.00.13
2	Menyampaikan keperluan	-	00.00.30	-	00.01.00	-	00.00.50	-	00.01.00	-	00.01.10	-
3	Menemui anggota Komisi III	-	00.05.00	-	00.01.00	-	00.10.00	-	00.00.30	-	00.01.00	-
4	Menyampaikan Aspirasi	Menyampaikan Aspirasi	00.05.00	00.02.31	00.05.00	00.02.05	00.05.00	00.02.10	00.05.00	00.01.43	00.05.00	00.02.20
5	-	Memantau perkembangan aspirasi	-	00.00.56	-	00.00.30	-	00.00.25	-	00.00.39	-	00.00.35
6	-	Memberikan Tanggapan	-	00.01.00	-	00.00.40	-	00.00.55	-	00.00.30	-	00.00.50
7	Mendapatkan hasil	Mendapatkan hasil	00.01.00	00.01.00	00.01.00	00.00.55	00.05.00	00.00.55	00.01.00	00.01.00	00.01.00	00.01.00

Tabel 7.21 Hasil perhitungan waktu yang dibutuhkan anggota komisi III

No	aktivitas yang dikerjakan		Waktu yang diperlukan			
			Responden 1		Responden 2	
	<i>As-Is</i>	<i>To-Be</i>	<i>As-Is</i>	<i>To-Be</i>	<i>As-Is</i>	<i>To-Be</i>
1	Menginformasikan agenda Safari fakultas	Mengakses Admin Sistem (SIAP)	00.02.00	00.00.10	00.02.00	00.00.10
2	Melaksanakan kegiatan Safari	-	01.00.00	-	01.00.00	-
3	Menemui mahasiswa/lembaga	-	00.30.00	-	00.15.00	-
4	Menanyakan & mencatat aspirasi	-	00.06.00	-	00.07.00	-
5	Mengklasifikasikan aspirasi	Mengklasifikasikan aspirasi	00.02.30	00.00.30	00.02.00	00.00.35
6	-	Mencetak Aspirasi	-	00.00.10	-	00.00.05
7	Menyampaikan ke pihak terkait	Menyampaikan ke pihak terkait	00.05.00	00.05.00	00.05.00	00.02.00
8	Menindaklanjuti perkembangan	Menindaklanjuti perkembangan	00.03.00	00.03.00	00.03.00	00.03.00
9	Mencatat hasil	Menyimpan perkembangan/hasil	00.05.00	00.07.00	00.05.00	00.08.00
10	Menyampaikan hasil	-	00.05.00	-	00.01.00	-

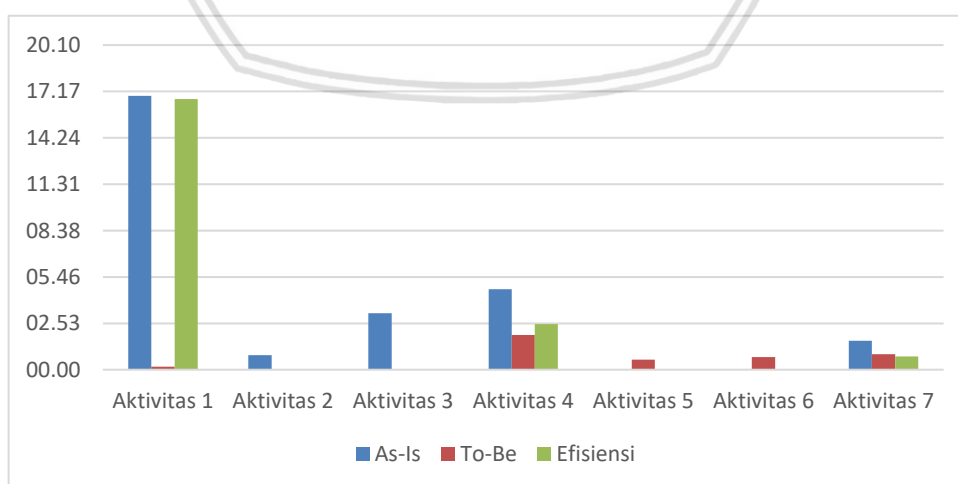
Hasil dalam Tabel 7.21 di atas menunjukkan waktu dari masing-masing aktivitas yang dilakukan dalam pemodelan proses bisnis *as-is* dan *to-be*. Dalam model *as-is* dimulai dari DPM UB yang menginformasikan untuk mengadakan agenda safari pada fakultas tertentu, lalu anggota Komisi III DPM UB akan datang ke fakultas untuk melaksanakan kegiatan tersebut. Dalam kegiatan tersebut mahasiswa/lembaga dengan anggota komisi III DPM UB akan terfasilitasi untuk saling bertemu, lalu anggota komisi III DPM UB dapat menanyakan dan mencatat aspirasi mahasiswa/lembaga. Aspirasi akan diklasifikasikan berdasarkan kategori dan tujuannya permasalahannya, lalu anggota DPM UB menyampaikan ke pihak terkait dan menindaklanjuti perkembangan secara berkala. Hasil yang didapatkan dari pihak terkait lalu dicatat dan disampaikan kepada pengaspirasi melalui telepon atau bertatap muka langsung. Sedangkan dalam model bisnis *to-be*, anggota komisi III DPM UB cukup mengakses admin sistem (SIAP) dan dapat mengkalifikasikan aspirasi menggunakan sistem lalu mencetak aspirasi untuk disampaikan ke pihak terkait. Dalam menindaklanjuti aspirasi tetap melakukan aktivitas yang sama namun dalam mencatat hasil anggota DPM UB dapat langsung menyimpan perkembangan atau hasil dari aspirasi yang didapatkan dari pihak terkait, dan secara otomatis sistem akan menyampaikan ke mahasiswa.

7.4.2 Analisis Hasil Pengujian Efisiensi Waktu

Data yang sudah didapatkan sebelumnya, selanjutnya dilakukan analisis perbandingan waktu dari masing-masing aktivitas pada proses bisnis yang telah dilakukan untuk mendapatkan nilai efisiensinya. Hasil analisis efisiensi waktu dalam menyampaikan aspirasi yang dilakukan oleh mahasiswa atau lembaga dipaparkan dalam Tabel 7.22 dan Gambar 7.4 berikut ini.

Tabel 7.22 Persentase efisiensi waktu mahasiswa atau lembaga

No	Aktivitas yang dikerjakan		Rata-rata waktu yang diperlukan responden		Efisiensi Waktu
	<i>As-Is</i>	<i>To-Be</i>	<i>As-Is</i>	<i>To-Be</i>	
1	Mendatangi sekretariat DPM UB	Mengakses SIAP Online DPM UB	00.17.00	00.00.12	98,8%
2	Menyampaikan keperluan	-	00.00.54	-	-
3	Menemui anggota Komisi III	-	00.03.30	-	-
4	Menyampaikan Aspirasi	Menyampaikan Aspirasi	00.05.00	00.02.10	56,7%
5	-	Memantau perkembangan aspirasi	-	00.00.37	-
6	-	Memberikan Tanggapan	-	00.00.47	-
7	Mendapatkan hasil	Mendapatkan hasil	00.01.48	00.00.58	46,3%
Total			00.28.12	00.04.44	83,2%

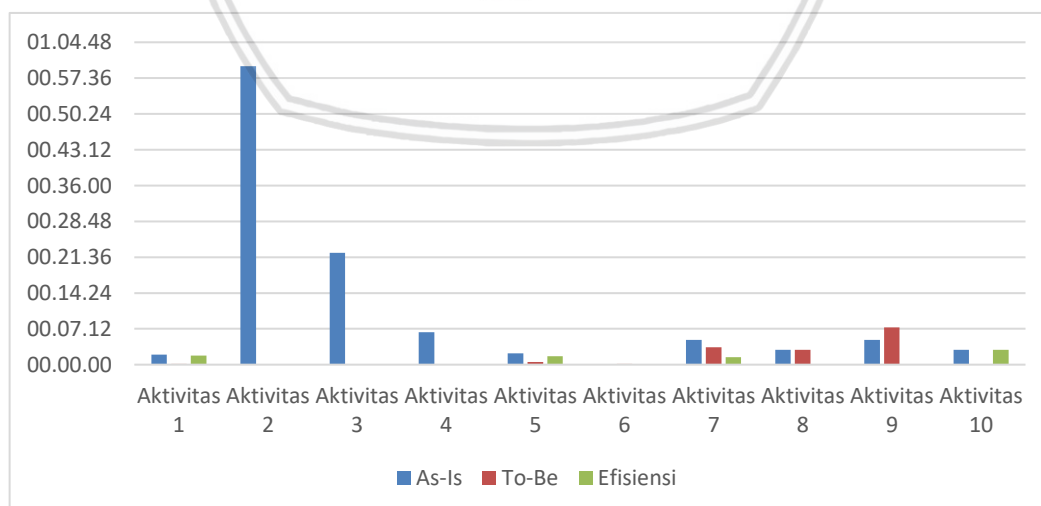


Gambar 7.4 Grafik Efisiensi Waktu Mahasiswa atau Lembaga

Sedangkan untuk analisis efisiensi waktu anggota komisi III DPM UB dalam mengelola aspirasi terdapat dalam Tabel 7.23 dan Gambar 7.5.

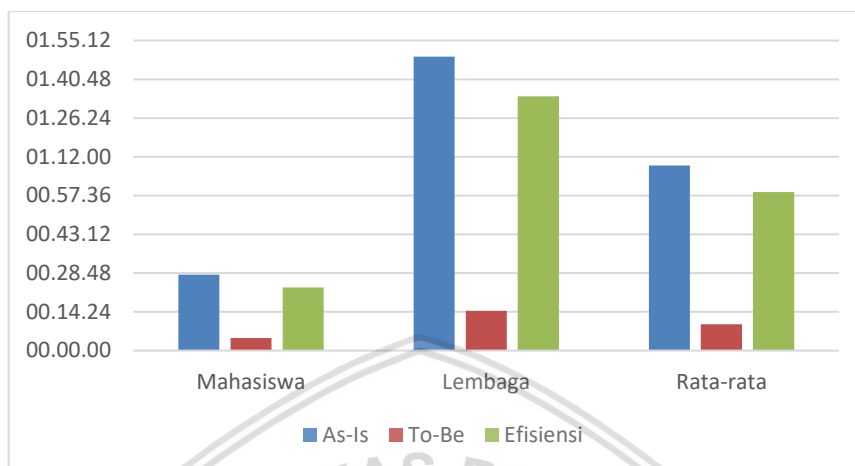
Tabel 7.23 Persentase efisiensi waktu anggota komisi III DPM UB

No	Aktivitas yang dikerjakan		Rata-rata waktu yang diperlukan responden		Efisiensi waktu
	<i>As-Is</i>	<i>To-Be</i>	<i>As-Is</i>	<i>To-Be</i>	
1	Menginformasikan agenda Safari fakultas	Mengakses Admin Sistem (SIAP)	00.02.00	00.00.10	91,7%
2	Melaksanakan kegiatan Safari	-	01.00.00	-	-
3	Menemui mahasiswa/lembaga	-	00.22.30	-	-
4	Menanyakan & mencatat aspirasi	-	00.06.30	-	-
5	Mengklasifikasikan aspirasi	Mengklasifikasikan aspirasi	00.02.15	00.00.33	75,9%
6	-	Mencetak Aspirasi	-	00.00.08	-
7	Menyampaikan ke pihak terkait	Menyampaikan ke pihak terkait	00.05.00	00.03.30	30%
8	Menindaklanjuti perkembangan	Menindaklanjuti perkembangan	00.03.00	00.03.00	0%
9	Mencatat hasil	Menyimpan perkembangan/hasil	00.05.00	00.07.30	-50%
10	Menyampaikan hasil	-	00.03.00	-	100%
Total			01.49.15	00.14.50	86,4%



Gambar 7.5 Grafik Efisiensi Waktu Anggota Komisi III DPM UB

Dari kedua hasil simulasi yang telah dilakukan sebelumnya dengan menggunakan skenario yang berbeda terhadap mahasiswa atau lembaga dan anggota komisi III DPM UB maka didapatkan rata-rata persentase dalam Gambar 7.6 berikut ini.



Gambar 7.6 Grafik Rata-rata hasil uji Efisiensi Waktu

Didapatkan rata-rata efisiensi waktu sebesar 84,4% dari nilai total efisiensi waktu bagi mahasiswa atau lembaga sebesar 83,2% dan anggota DPM UB sebesar 86,4%. Hal ini menunjukkan bahwa nilai dari efisiensi waktu proses bisnis usulan (*to-be*) terhadap proses bisnis saat ini (*as-is*) lebih baik karena dapat meminimalisir waktu sebesar 84,4%.

BAB 8 PENUTUP

Bab ini memuat laporan akhir dari hasil penelitian yang telah dilakukan, terdiri dari kesimpulan dan saran untuk penelitian selanjutnya. Kesimpulan memuat ringkasan hasil dari penelitian, sedangkan saran didapatkan dari umpan balik atau masukan dari pengguna sistem serta dari peneliti.

8.1 Kesimpulan

Berdasarkan hasil pemodelan proses bisnis, analisis persyaratan, analisis komponen, modifikasi persyaratan, perancangan menggunakan komponen, implementasi dan integrasi, serta pengujian dan analisis yang telah dilakukan dalam penelitian ini, maka didapatkan kesimpulan sebagai berikut:

1. Hasil 3 jenis pemodelan proses bisnis pada pelayanan aspirasi di DPM UB, didapatkan usulan pembaruan pemodelan proses bisnis. Dari pemodelan *as-is* dan *to-be* didapatkan perubahan aktivitas pelayanan yaitu: mengeliminasi 8 aktivitas pada proses bisnis saat ini, mengubah 5 aktivitas yang sebelumnya dilakukan secara manual menjadi menggunakan sistem atau otomatis dilakukan oleh sistem, dan menambahkan 4 aktivitas baru pada pemodelan proses bisnis usulan. Sehingga berdasarkan hasil uji efisensi waktu didapatkan efektifitas dan efisiensi pemodelan proses bisnis yang lebih baik.
2. Hasil analisis persyaratan merupakan dokumentasi yang memuat informasi berikut ini:
 - a. Hasil elisitasi persyaratan yang teridentifikasi 4 tipe pemangku kepentingan, 2 pernyataan permasalahan pemangku kepentingan, 9 kebutuhan pemangku kepentingan, 4 tipe pengguna sistem, dan 11 fitur yang dapat menjadi solusi kebutuhan pengguna.
 - b. Hasil spesifikasi persyaratan sistem yang memuat 17 persyaratan fungsional sistem dan 4 persyaratan non-fungsional sistem.
 - c. Hasil verifikasi dan validasi persyaratan fungsional dan non-fungsional sistem kepada DPM UB selaku pihak yang berwenang dalam pemangku kepentingan.
 - d. Hasil pemodelan *use case* teridentifikasi 5 aktor yang terlibat terhadap sistem, 10 *use case* atau interaksi aktor terhadap sistem beserta spesifikasinya.
3. Hasil analisis rancangan dan implementasi sistem menggunakan pemodelan *reuse-oriented development* didapatkan hasil sebagai berikut:
 - a. Rancangan sistem berdasarkan hasil analisis komponen, modifikasi persyaratan dan perancangan dengan komponen didapatkan 8 komponen *reuse* yang digunakan ulang ke dalam proses implementasi sistem, 2 komponen yang dilakukan modifikasi agar memenuhi persyaratan, dan 6 jenis desain rancangan sistem yang meliputi: rancangan arsitektur sistem, beberapa rancangan interaksi objek dalam bentuk *sequence diagram*, 3 desain diagram rancangan kelas yang memuat

- 10 kelas beserta relasinya, 11 rancangan tabel basis data beserta relasinya, beberapa rancangan algoritme fungsi sistem, dan 7 rancangan antarmuka sistem.
- b. Hasil implementasi sistem memuat spesifikasi lingkungan implementasi, implementasi serta integrasi algoritma pada beberapa fungsi yang meliputi: fungsi login mahasiswa, login lembaga, mengirimkan aspirasi, memantau aspirasi, memberikan tanggapan dan mengelola akun. Yang mana dalam proses pengembangannya tidak memerlukan implementasi dari nol, karena menggunakan pemodelan *reuse-oriented* sehingga menjadi lebih cepat. Dalam hal ini juga terdapat hasil implementasi antarmuka sistem meliputi: antarmuka halaman utama, login admin, profil, kirim aspirasi, statistik aspirasi, kelola data aspirasi, detail aspirasi, cetak aspirasi dan kelola akun.
4. Hasil uji dari 4 jenis pengujian yang dilakukan pada Sistem Informasi Aspirasi Online pada DPM UB, didapatkan hasil sebagai berikut:
- a. Pengujian validasi menggunakan metode *black-box* yang dilakukan pada masing-masing *use case* didapatkan hasil 100% valid dan dapat disimpulkan bahwa sistem sudah memenuhi persyaratan fungsional dan siap dirilis kepada pengguna akhir.
 - b. Pengujian kompatibilitas yang dilakukan pada persyaratan non-fungsional terhadap *compatibility* dan *accessibility* sistem mendapatkan skor rata-rata 89% sehingga dapat kesimpulan bahwa persyaratan non-fungsional tersebut sudah sangat baik.
 - c. Pengujian penerimaan pengguna yang dilakukan pada 26 responden untuk mengetahui *usability* dan *security* sistem berdasarkan penilaian pengguna mendapatkan hasil rata-rata keseluruhan sebesar 81%, sehingga dapat disimpulkan bahwa sistem dapat diterima oleh pengguna akhir untuk digunakan dalam melayani aspirasi pada DPM UB.
 - d. Pengujian efisiensi waktu yang dilakukan untuk mengetahui efisiensi waktu model proses bisnis *to-be* terhadap *as-is* mendapatkan rata-rata keseluruhan efisiensi waktu sebesar 84,4%.

8.2 Saran

Saran-saran yang diberikan untuk melakukan pengembangan selanjutnya, meliputi:

1. Perlu ditambahkan panduan penggunaan sistem, penjelasan dari masing-masing fungsi sistem, peningkatan keamanan dan kenyamanan terhadap privasi dari informasi yang digunakan, serta penggunaan grafik/gambar yang lebih baik.
2. Perlu ditambahkan fitur pelayanan aspirasi menggunakan SMS *gateway*, baik dalam proses penyampaiannya maupun pemberitahuan dari perkembangan aspirasi yang disampaikan.
3. Perlu dilakukan evaluasi terhadap sistem untuk mengetahui kepuasan pengguna terhadap pelayanan aspirasi yang diberikan.

DAFTAR PUSTAKA

- Baldassarre, M. T., Bianchi, A., Caivano, D. & Visaggio, G., 2005. *An Industrial Case Study on Reuse Oriented Development*.
- Becker, J., Kugeler, M. & Rosemann, M., 2013. *Process Management: A Guide for the Design of Business Processes*. Berlin: Springer-Verlag.
- Bittner, K. & Spence, I., 2002. *Use Case Modeling*. s.l.:Addison Wesley.
- Buttle, F., 2009. *Customer Relationship Management. Second Edition*. [Online] Available at: <http://books.google.com> [Diakses 12 Februari 2018].
- Guru99, 2018. *Compatibility Testing Tutorial: Forward & Backward Testing*. [Online] Available at: <https://www.guru99.com/compatibility-testing.html> [Diakses 10 April 2018].
- Hendryadi & Suryani, 2015. *Metode Riset Kuantitatif*. Jakarta: Prenadamedia Group.
- Hurlock, E. B., 1979. *Personality Development*. New Delhi: Tata McGraw-Hill Publishing Company Ltd.
- IBM, 2007. *(RR631) Writing Good Use Cases*. Cupertino, USA: Student Workbook. IBM Rational University.
- KBBI, 2018. *Kamus Besar Bahasa Indonesia (KBBI)*. [Online] Available at: <https://kbbi.web.id/aspirasi> [Diakses 12 Februari 2018].
- Kurniawan, T. A., 2018. *Pemodelan Use Case (UML): Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik*, Volume 5, pp. 77-86.
- Lau, J., 2012. *2012T1 Fortune Cookies: User Testing*. [Online] Available at: https://wiki.smu.edu.sg/is480/2012T1_Fortune_Cookies%3A_User_Testing [Diakses 10 April 2018].
- Nielsen, J., 2012. *Usability 101: Introduction to Usability*. [Online] Available at: <https://www.nngroup.com/articles/usability-101-introduction-to-usability> [Accessed 25 Agustus 2018].
- Pressman, R. S., 2010. *Software Engineering: A Practitioner's Approach*. 7th ed. New York: McGraw-Hill Book Company.
- Prihantoro, W., Munadi, R. & Iqbal, M., 2011. *Perancangan Dan Implementasi Sistem Informasi Dan Aplikasi*.

- Przybyłek, A., 2011. *Bridging the gap between business process models and use-case models*. Gdańsk, Gdańsk University of Technology.
- Septian, G., 2011. *Trik Pintar Menguasai Codeigniter*. Jakarta: PT Elex Media Komputindo.
- Siregar, M. R. et al., 2016. *Implementasi Good Governance pada Organisasi Mahasiswa di Universitas Gadjah Mada Demi Mewujudkan Good Student Governance*.
- Sommerville, I., 2011. *Software Engineering*. 9th ed. America: Pearson Education, Inc.
- Sutoyo, A., 2009. *Pemahaman Individu, Observasi, Checklist, Interviu, Kuesioner dan Sosiometri*. Yogyakarta: Pustaka Pelajar.
- Tahir, M. et al., 2016. *Framework for Better Reusability in Component Based Software Engineering*, pp. 77-81.
- Weske, M., 2007. *Business Process Management Concepts, Languages, Architectures*. Postdam: Springer-Verlag Berlin Heidelberg.

